

## [EP] Общее описание проекта

Ссылка на задачу Jira

Дочерние страницы

- [\[EP\] 1. Роли пользователей](#)
- [\[EP\] Черновик](#)


### Оглавление

[Дочерние страницы](#)  
[Общее описание](#)  
[API-документация Внешней системы](#)  
[Ограничения Внешней системы:](#)  
[Сценарий работы \(Use Case\)](#)  
[Техническая реализация метода](#)  
[Запрос](#)  
[Ответ](#)  
[Успешный ответ](#)  
[Требования к обработке ошибок](#)  
[Маппинг данных](#)  
[Маппинг данных UI - API нашей системы](#)  
[Название экрана](#)  
[Логирование](#)  
[Дополнительно](#)

## Общее описание

 Общее описание задачи, которую необходимо реализовать.

- Что это?
- Зачем? (какие возможности появятся)
- Почему? (какие проблемы решаем?)
- Для кого?
- Сколько стоит?
- Для чего? Почему мы это делаем?
- Скриншоты или макеты, для быстрого понимания о чем речь (если это важно)

 Дополнительно сюда можно добавить схему архитектуры, касающуюся этой части системы.

## API-документация Внешней системы



Ссылка на API-документацию, PDF или другие материалы, которые нужны разработчику для работы над задачей. Можно добавить сюда копию методов, но лучше ссылаться на первоисточник.

#### Ограничения Внешней системы:

- [-] Нефункциональные требования к интеграции:
  - время обработки запросов внешней системой,
  - лимит на количество запросов в секунду / минуту / час и т.д,
  - другие ограничения.

## Сценарий работы (Use Case)

#### Системы:

- [-] Компоненты (сервисы, микросервисы, мобильные и веб-приложения, внешние системы), участвующие в сценарии интеграции

#### Входные данные:

- [-] Список данных на вход

#### Предусловие:

- [-] В каком состоянии система?  
Что инициирует старт работы сценария?

#### Алгоритм работы:

- [-] Логика работы системы.  
В интеграционном Use Case обязательно указывать какие конкретно методы API вызываются при работе метода.

#### Альтернативные сценарии и обработка ошибок:


- [-] Требования к обработке ошибок по каждому шагу.

#### Результат:

- [-] Данные, которые формируются в результате работы метода.

- [-] Дополнительно можно добавить к сценарию UML Sequence диаграмму.

## Техническая реализация метода

 Не обязательно, но желательно заполнять этот раздел.  
Могут заполнять разработчики.

Пример приведен для интеграционного REST API метода.

### Запрос

Тип метода:

URL:

Headers:

- 

Авторизация:

- 

Query-параметры:

- 

Тело запроса:

1

### Ответ

Успешный ответ

Код: HTTP-200/201

Тело ответа:

1

Требования к обработке ошибок


1. Код: HTTP-4\*\*/5\*\*

 Описание

Тело ответа:

1


2. Код: HTTP-4\*\*/5\*\*

 Описание

Тело ответа:

1

Маппинг данных

 Основная система - для которой  
постановка задачи

Запрос:

Параметр	В ответе JSON Основной системы	В JSON от Внешней системы	В БД Основ ной системы	Тип данных в БД	Коммента рий

Ответ:

Параметр	В ответе JSON Основной системы	В JSON от Внешней системы	В БД Основ ной системы	Тип данных в БД	Коммента рий

Маппинг данных UI - API нашей системы

Название экрана

Описание

Ссылка на макет: ...

Параметр на UI	В ответе JSON Основной системы (U-Verse)	Описание и требования к ФЛК (форматно-логическому контролю)

## Логирование

- ☐ Требования к логированию и мониторингу взаимодействия систем.

## Дополнительно

- ☐ В состав постановки задач могут быть включены:
  - глоссарий (предпочтительно вести общий, по всей системе, но зависит от удобства для команды),
  - требования к доработке БД (предпочтительно делать отдельно, но зависит от удобства для команды),
  - требования к аудиту (отслеживанию действий пользователя), если такая функциональность есть в системе.