

# Описание

Редактирования метаданных

## Доступ

- роль META\_ADMIN

## Настройки редактора

Редактор меты имеет по умолчанию код `metastorage` , который переопределяется настройкой `jprime.metaEditor.code`

## Доступ по коду сервиса

При наличии настройки `jprime.application.code` все запросы также доступны по маске `/dynamicmeta/v1/{jprime.application.code}/...`

## Информация о конструкторе меты

### 1. Получение списка конструкторов

GET `/dynamicmeta/v1/apps/`

- ответ

```
{
  "apps": [
    {
      "code": "reportseditor",
      "title": "Конструктор отчетов"
    }
  ]
}
```

Параметр	Описание
apps	Список конструкторов
code	Код конструктора
title	Название конструктора

## Операции с метаописанием

### 1. Получение списка хранилищ

GET `/dynamicmeta/v1/metaStorages/`

- ответ

Код	Описание
401	Нет доступа
404	Адрес не найден
500	Ошибка сервера
200	Логика отработала успешно

```
[
  {
    "code": "metastorage",
    "title": "Хранилище метаданных"
  },
  {
    "code": "metaFileStorage",
    "title": "Хранилище данных о файлах"
  },
  {
    "code": "main",
    "title": "Основное хранилище данных"
  }
]
```

Параметр	Описание
code	Код типа хранилища
title	Название хранилища

Список хранилищ составляется на основе настроек:

`jpri.me.metaEditor.storageFilter.enabled` - признак работы фильтра

`jpri.me.metaEditor.storageFilter.jpStorageCodes` - список разрешённых кодовых имен хранилищ через []

## 2. Получение регистра значений

GET /dynamicmeta/v1/jpCases/

- ответ

Код	Описание
401	Нет доступа
404	Адрес не найден
500	Ошибка сервера

200 Код	Логика отработала успешно Описание
------------	---------------------------------------

```
[
  {
    "code": "upper",
    "title": "Верхний регистр"
  },
  {
    "code": "lower",
    "title": "Нижний регистр"
  },
  {
    "code": "any",
    "title": "Произвольный (по умолчанию)"
  }
]
```

Параметр	Описание
code	Код регистра
title	Название

### 3. Создание меты с указанием маппинга

POST /dynamicmeta/v1/jpClasses/

- запрос

```
{
  "code": "testClass",
  "qName": "common.testClass",
  "name": "Тестовый класс 2",
  "jpPackage": "adminAccess",
  "attrs": [
    {
      "code": "id1",
      "qName": "common.testClass.id1",
      "name": "id",
      "identifier": true,
      "mandatory": true,
      "type": "string",
      "length": 0
    },
    {
      "code": "id2",
      "qName": "common.testClass.id2",
      "name": "id",
      "type": "string",
      "length": 0
    }
  ]
}
```

```

    },
    {
        "code": "id3",
        "qName": "common.testClass.id3",
        "name": "id",
        "type": "json",
        "length": 0,
        "jpProps": [
            {
                "code": "testprop",
                "qName": "egu.eguApplication.jsonattr.jsonprop",
                "name": "Test Prop",
                "mandatory": false,
                "multiple": false,
                "type": "element",
                "length": null,
                "refJpClassCode": null,
                "refJpAttrCode": null,
                "jpProps": [
                    {
                        "code": "innertestprop",
                        "qName": "egu.eguApplication.jsonattr.injsonprop",
                        "name": "Inner Test Prop",
                        "shortName": null,
                        "description": null,
                        "mandatory": false,
                        "multiple": false,
                        "type": "string",
                        "length": null,
                        "refJpClassCode": null,
                        "refJpAttrCode": null,
                        "jpProps": []
                    }
                ]
            }
        ]
    }
],
"maps": [
    {
        "storage": "main",
        "map": "rx_testclass",
        "attrMaps": [
            {
                "map": "id1",
                "code": "id1"
            },
            {
                "map": "id2",
                "code": "id2"
            }
        ]
    }
]

```

```
    ]
  }
]
}
```

- ответ

Код	Описание
401	Нет доступа
404	Адрес не найден
500	Ошибка сервера
200	Логика отработала успешно

4. Обновление меты с указанием маппинга

```
PUT /dynamicmeta/v1/jpClasses/
```

- запрос

```

{
  "guid": "406824f9-0f30-40ba-bf2f-25fd0a838dac",
  "code": "testClass",
  "qName": "common.testClass",
  "name": "Тестовый класс 2",
  "jpPackage": "adminAccess",
  "version": "MjAyMi0wMS0xMFQxMzowNzoyNC4yNTI=",
  "attrs": [
    {
      "guid": "39c17f7f-c5b6-4ec4-8cdb-8ae8bbc1b353",
      "code": "id1",
      "qName": "common.testClass.id1",
      "name": "id",
      "identifier": true,
      "mandatory": true,
      "type": "string",
      "length": 0
    },
    {
      "guid": "5f5dfe2e-e2da-4bf4-ab67-8fb9286c3f7a",
      "code": "id2",
      "qName": "common.testClass.id2",
      "name": "id",
      "type": "string",
      "length": 0
    }
  ],
  "maps": [
    {
      "guid": "8abd3726-dc64-4d52-a5b6-d294b957a13d",
      "storage": "main",
      "map": "rx_testclass",
      "attrMaps": [
        {
          "guid": "4a84b593-dd8f-47de-815a-efb3834e98ed",
          "map": "id1",
          "code": "id1"
        },
        {
          "guid": "168a7bd5-2134-4bb8-b576-b70d6941ac9c",
          "map": "id2",
          "code": "id2"
        }
      ]
    }
  ]
}

```

Код	Описание
401	Нет доступа
404	Адрес не найден
500	Ошибка сервера
200	Логика отработала успешно

5. Чтение меты с указанием маппинга

GET /dynamicmeta/v1/jpClasses/

- ответ

Код	Описание
401	Нет доступа
404	Адрес не найден
500	Ошибка сервера
200	Логика отработала успешно

```
{
  "totalCount": 1,
  "count": 1,
  "classes": [
    {
      "guid": "406824f9-0f30-40ba-bf2f-25fd0a838dac",
      "code": "testClass",
      "qName": "common.testClass",
      "name": "Тестовый класс 2",
      "jpPackage": "adminAccess",
      "version": "MjAyMi0wMS0xMFQxMzowNzoyNC4yNTI=",
      "attrs": [
        {
          "guid": "39c17f7f-c5b6-4ec4-8cdb-8ae8bbc1b353",
          "code": "id1",
          "qName": "common.testClass.id1",
          "name": "id",
          "identifier": true,
          "mandatory": true,
          "type": "string",
          "length": 0,
          "jpProps": []
        },
        {
          "guid": "5f5dfe2e-e2da-4bf4-ab67-8fb9286c3f7a"
```

```

        "guid": "515d1e2e-e2da-4b14-a007-81b9280c317a",
        "code": "id2",
        "qName": "common.testClass.id2",
        "name": "id",
        "type": "string",
        "length": 0,
        "jpProps": []
    }
],
"maps": [
    {
        "guid": "8abd3726-dc64-4d52-a5b6-d294b957a13d",
        "storage": "main",
        "map": "rx_testclass",
        "attrMaps": [
            {
                "guid": "4a84b593-dd8f-47de-815a-efb3834e98ed",
                "map": "id1",
                "code": "id1"
            },
            {
                "guid": "168a7bd5-2134-4bb8-b576-b70d6941ac9c",
                "map": "id2",
                "code": "id2"
            }
        ]
    }
]
}
]
}
]
}
}

```

## 6. Поиск меты

POST /dynamicmeta/v1/jpClasses/search

по общим правилам

### Форматы запроса

- название содержит



```
{
  "offset": 0,
  "limit": 50,
  "filter": {
    "cond": {
      "attr": "longName",
      "like": "МАЯК"
    }
  }
}
```

- ответ

Код	Описание
401	Нет доступа
404	Адрес не найден
500	Ошибка сервера
200	Логика отработала успешно

## 7. Чтение определенной меты с указанием маппинга

```
GET /dynamicmeta/v1/jpClasses/<jpClassGuid>
```

- ответ

Код	Описание
401	Нет доступа
404	Адрес не найден
500	Ошибка сервера
200	Логика отработала успешно

```

{
  "guid": "406824f9-0f30-40ba-bf2f-25fd0a838dac",
  "code": "testClass",
  "qName": "common.testClass",
  "name": "Тестовый класс 2",
  "jpPackage": "adminAccess",
  "version": "MjAyMi0wMS0xMFQxMzowNzoyNC4yNTI=",
  "attrs": [
    {
      "guid": "39c17f7f-c5b6-4ec4-8cdb-8ae8bbc1b353",
      "code": "id1",
      "qName": "common.testClass.id1",
      "name": "id",
      "identifier": true,
      "mandatory": true,
      "type": "string",
      "length": 0
    },
    {
      "guid": "5f5dfe2e-e2da-4bf4-ab67-8fb9286c3f7a",
      "code": "id2",
      "qName": "common.testClass.id2",
      "name": "id",
      "type": "string",
      "length": 0
    }
  ],
  "maps": [
    {
      "guid": "8abd3726-dc64-4d52-a5b6-d294b957a13d",
      "storage": "main",
      "map": "rx_testclass",
      "attrMaps": [
        {
          "guid": "4a84b593-dd8f-47de-815a-efb3834e98ed",
          "map": "id1",
          "code": "id1"
        },
        {
          "guid": "168a7bd5-2134-4bb8-b576-b70d6941ac9c",
          "map": "id2",
          "code": "id2"
        }
      ]
    }
  ]
}

```

## 8. Удаление определенной меты с указанием маппинга

DELETE /dynamicmeta/v1/jpClasses/<jpClassGuid>

- ответ

Код	Описание
401	Нет доступа
404	Адрес не найден
500	Ошибка сервера
202	Логика отработала успешно

## 9.1. Инициализация меты

POST dynamicmeta/v1/dynamicmeta-metainit/mode/init

- ответ

Код	Описание
401	Нет доступа
404	Адрес не найден
500	Ошибка сервера
200	Логика отработала успешно

```
{
  "description": "Метаописание проинициализировано",
  "changeData": false,
  "resultType": "message",
  "qName": "dynamicmeta-metainit.init.ready"
}
```

## 9.2. Инициализация меты по идентификаторам

POST dynamicmeta/v1/dynamicmeta-metainit/mode/initByIds

- запрос

```
{
  "objectIds": [
    "b01fabbb-0765-46b2-9c6b-67b64f665725"
  ]
}
```

- ответ

Код	Описание
401	Нет доступа
404	Адрес не найден
500	Ошибка сервера
200	Логика отработала успешно

```
{
  "description": "Метаописание проинициализировано",
  "changeData": false,
  "resultType": "message",
  "qName": "dynamicmeta-metainit.initByIds.ready"
}
```

9.3. Инициализация меты по данным авторизации

POST dynamicmeta/v1/dynamicmeta-metainit/mode/userInit

- ответ

Код	Описание
401	Нет доступа
404	Адрес не найден
500	Ошибка сервера
200	Логика отработала успешно

```
{
  "description": "Метаописание проинициализировано",
  "changeData": false,
  "resultType": "message",
  "qName": "dynamicmeta-metainit.userInit.ready"
}
```

10. Синхронизация структуры метаописания и хранилища

POST dynamicmeta/v1/dynamicmeta-storagesync/mode/sync

- ответ

Код	Описание
401	Нет доступа

Код	Описание
404	Адрес не найден
500	Ошибка сервера
200	Логика отработала успешно

```
{
  "description": "Структура данных синхронизирована",
  "changeData": false,
  "resultType": "message",
  "qName": "dynamicmeta-storagesync.init.ready"
}
```

11. Синхронизация структуры метаописания и хранилища определенного класса

POST dynamicmeta/v1/dynamicmeta-storagesync/mode/classsync

- запрос

```
{
  "guid": "406824f9-0f30-40ba-bf2f-25fd0a838dac"
}
```

Код	Описание
guid	Гуид динамического класса для инициализации

- ответ

Код	Описание
401	Нет доступа
404	Адрес не найден
500	Ошибка сервера
200	Логика отработала успешно

```
{
  "description": "Структура данных синхронизирована",
  "changeData": false,
  "resultType": "message",
  "qName": "dynamicmeta-storagesync.init.ready"
}
```

Операции с шаблоном меты

## 1. Создание шаблона меты

POST /dynamicmeta/v1/jpPatterns/

- запрос

```
{
  "code": "testClass",
  "name": "Тестовый класс 2",
  "jpPackage": "adminAccess",
  "number": 1,
  "attrs": [
    {
      "code": "id1",
      "qName": "common.testClass.id1",
      "name": "id",
      "identifier": true,
      "mandatory": true,
      "type": "string",
      "length": 0
    },
    {
      "code": "id2",
      "qName": "common.testClass.id2",
      "name": "id",
      "type": "string",
      "length": 0
    },
    {
      "code": "id3",
      "qName": "common.testClass.id3",
      "name": "id",
      "type": "json",
      "length": 0,
      "jpProps": [
        {
          "code": "testprop",
          "qName": "egu.eguApplication.jsonattr.jsonprop",
          "name": "Test Prop",
          "mandatory": false,
          "multiple": false,
          "type": "element",
          "length": null,
          "refJpClassCode": null,
          "refJpAttrCode": null,
          "jpProps": [
            {
              "code": "innertestprop",
              "qName": "egu.eguApplication.jsonattr.injsonprop",
              "name": "Inner Test Prop",
              "shortName": null,
              "description": null,
```

```
        "mandatory": false,
        "multiple": false,
        "type": "string",
        "length": null,
        "refJpClassCode": null,
        "refJpAttrCode": null,
        "jpProps": []
      }
    ]
  }
]
}
```

- ответ

Код	Описание
401	Нет доступа
404	Адрес не найден
500	Ошибка сервера
200	Логика отработала успешно

## 2. Обновление шаблона меты

```
PUT /dynamicmeta/v1/jpPatterns/
```

- запрос

```
{
  "guid": "406824f9-0f30-40ba-bf2f-25fd0a838dac",
  "code": "testClass",
  "name": "Тестовый класс 2",
  "jpPackage": "adminAccess",
  "number": 1,
  "attrs": [
    {
      "guid": "39c17f7f-c5b6-4ec4-8cdb-8ae8bbc1b353",
      "code": "id1",
      "qName": "common.testClass.id1",
      "name": "id",
      "identifier": true,
      "mandatory": true,
      "type": "string",
      "length": 0
    },
    {
      "guid": "5f5dfe2e-e2da-4bf4-ab67-8fb9286c3f7a",
      "code": "id2",
      "qName": "common.testClass.id2",
      "name": "id",
      "type": "string",
      "length": 0
    }
  ]
}
```

- ответ

Код	Описание
401	Нет доступа
404	Адрес не найден
500	Ошибка сервера
200	Логика отработала успешно

### 3. Чтение шаблона меты

GET /dynamicmeta/v1/jpPatterns/

- ответ

Код	Описание
401	Нет доступа
404	Адрес не найден



Код	Описание сервера
200	Логика отработала успешно

```
{
  "totalCount": 1,
  "count": 1,
  "classes": [
    {
      "guid": "406824f9-0f30-40ba-bf2f-25fd0a838dac",
      "code": "testClass",
      "name": "Тестовый класс 2",
      "jpPackage": "adminAccess",
      "number": 1,
      "attrs": [
        {
          "guid": "39c17f7f-c5b6-4ec4-8cdb-8ae8bbc1b353",
          "code": "id1",
          "qName": "common.testClass.id1",
          "name": "id",
          "identifier": true,
          "mandatory": true,
          "type": "string",
          "length": 0,
          "jpProps": []
        },
        {
          "guid": "5f5dfe2e-e2da-4bf4-ab67-8fb9286c3f7a",
          "code": "id2",
          "qName": "common.testClass.id2",
          "name": "id",
          "type": "string",
          "length": 0,
          "jpProps": []
        }
      ]
    }
  ]
}
```

## 4. Поиск шаблона меты

POST /dynamicmeta/v1/jpPatterns/search

по общим правилам

### Форматы запроса

- название содержит

```
{
  "offset": 0,
  "limit": 50,
  "filter": {
    "cond": {
      "attr": "longName",
      "like": "МАЯК"
    }
  }
}
```

- ответ

Код	Описание
401	Нет доступа
404	Адрес не найден
500	Ошибка сервера
200	Логика отработала успешно

## 5. Чтение определенного шаблона меты

GET /dynamicmeta/v1/jpPatterns/<patternGuid>

- ответ

Код	Описание
401	Нет доступа
404	Адрес не найден
500	Ошибка сервера
200	Логика отработала успешно

```
{
  "guid": "406824f9-0f30-40ba-bf2f-25fd0a838dac",
  "code": "testClass",
  "name": "Тестовый класс 2",
  "jpPackage": "adminAccess",
  "number": 1,
  "attrs": [
    {
      "guid": "39c17f7f-c5b6-4ec4-8cdb-8ae8bbc1b353",
      "code": "id1",
      "qName": "common.testClass.id1",
      "name": "id",
      "identifier": true,
      "mandatory": true,
      "type": "string",
      "length": 0
    },
    {
      "guid": "5f5dfe2e-e2da-4bf4-ab67-8fb9286c3f7a",
      "code": "id2",
      "qName": "common.testClass.id2",
      "name": "id",
      "type": "string",
      "length": 0
    }
  ]
}
```

## 8. Удаление определенной меты с указанием маппинга

DELETE /dynamicmeta/v1/jpPatterns/<patternGuid>

- ответ

Код	Описание
401	Нет доступа
404	Адрес не найден
500	Ошибка сервера
202	Логика отработала успешно

## 9. Получение свойств типа атрибута

GET /dynamicmeta/v1/attrTypeProperties/

- ответ

```
[
  {
    "code": "backReference",
    "availableChanges": []
  },
  {
    "code": "biginteger",
    "availableChanges": [
      "string",
      "simpleFraction",
      "money",
      "file"
    ]
  },
  ...
]
```

Параметр	Описание
code	Код типа атрибута
availableChanges	Список типов, на которые можно изменить текущий

## 10. Добавление обратной ссылки в уже имеющийся класс

POST /dynamicmeta/v1/backRefAttr/

- запрос

```
{
  "jpClassGuid": "4e301047-5494-4f00-bca2-1f9e165a9379",
  "code": "professorsScheduleRef2",
  "qName": "jp.sandbox.disciplinesNeat.professorsScheduleRef",
  "name": "Обратная ссылка на расписание преподавателя",
  "shortName": null,
  "description": null,
  "jpPackage": null,
  "refJpClass": "professorsScheduleNeat",
  "refJpAttr": "disciplinesId"
}
```

Параметр	Описание
jpClassGuid	Код класса, в который добавляется атрибут
code	Кодовое имя атрибута
qName	QName атрибута

name Параметр	имя атрибута Описание
shortName	короткое имя атрибута
description	описание атрибута
jpPackage	Настройки доступа к атрибуту
refJpClass	Класс-ссылка
refJpAttr	Атрибут, по которому строится обратная ссылка

- ответ (полное описание класса)

```
{
  "guid": "4e301047-5494-4f00-bca2-1f9e165a9379",
  "code": "testClass",
  ...
}
```

# Операции с файловым хранилищем

## 1. Получение списка файловых хранилищ

```
GET /dynamicmeta/v1/fileStorages/
```

- ответ

Код	Описание
401	Нет доступа
404	Адрес не найден
500	Ошибка сервера
200	Логика отработала успешно

```
[
  {
    "code": "metaFileStorage",
    "title": "Хранилище данных о файлах"
  },
  {
    "code": "main",
    "title": "Основное хранилище данных"
  }
]
```

Параметр	Описание
code	Код типа хранилища
title	Название хранилища

## 2. Описание проверки имени хранилища файлов

GET /dynamicmeta/v1/fileStorage/info

Ответ вида

```
[
  {
    "desc": "fist rule",
    "rules": []
  },
  {
    "desc": "second rule",
    "rules": [
      {
        "desc": "third rule desc second rule",
        "rules": []
      }
    ]
  }
]
```

## 3. Проверка имени хранилища файлов

POST /dynamicmeta/v1/fileStorage/check

Запрос вида

```
{
  "fileStorage": "storage-name"
}
```

ответ

- ответ

Код	Описание
400	Неверный запрос
404	Адрес не найден
500	Ошибка сервера
200	Проверка пройдена

Настройки

Код	Описание
fileStorageValidation.enabled	Включение сервиса валидации имени файлового хранилища
fileStorageValidation.rules	JSON регулярного выражения и описания правил валидации

Описание настроек

Настройки валидации имени хранилища файлов ( fileStorageValidation.rules )

Содержит строку с экранированием спецсимволов в виде

```
{
  "regex": "(?=^.{3,63}$)(?!^(\\d+\\.)+\\d+$)(^([a-z0-9]|[a-z0-9][a-z0-9\\-]*[a-z0-9])\\.)*([a-z0-9]|[a-z0-9][a-z0-9\\-]*[a-z0-9])$",
  "rules": [
    {
      "desc": "длина от 3 до 63 символов;"
    },
    {
      "desc": "содержание из маленьких латинских букв, цифр, точек(.) и тире(-);"
    },
    {
      "desc": "начало и окончание маленькая латинская буква или цифра;"
    },
    {
      "desc": "не более одной точки(.) подряд;"
    },
    {
      "desc": "не похоже на IP адрес(например: 192.168.5.4);"
    }
  ]
}
```

Пример с экранированием:

```
"{"regex":"(?!^(\\\\d+\\\\.)+\\\\d+$)(^([a-z0-9]|[a-z0-9][a-z0-9\\\\\\\\-]*[a-z0-9])\\\\.)*([a-z0-9]|[a-z0-9][a-z0-9\\\\\\\\-]*[a-z0-9])$","rules":[{"desc":"длина от 3 до 63 символов;\""}, {"desc":"содержание из маленьких латинских букв, цифр, точек(.) и тире(-);\""}, {"desc":"начало и окончание маленькая латинская буква или цифра;\""}, {"desc":"не более одной точки(.) подряд;\""}, {"desc":"не похоже на IP адрес(например: 192.168.5.4);\""}]"}"
```

Код	Описание

regex Код	Регулярное выражение проверки имени хранилища файлов Описание
desc	Описание настройки
rules	Вложенные настройки