



119270, Москва, Лужнецкая наб., д. 6,  
стр.1, офис 214, ООО «ЭР СИ О»  
Тел: (495) 287-98-87  
E-mail: [info@rco.ru](mailto:info@rco.ru)  
<http://www.rco.ru>

## **Руководство разработчика RCO Zoom SDK**

Москва, 2021

В содержание данного документа могут быть внесены изменения без предварительного уведомления. Названия организаций, имена и даты, используемые в качестве примеров, являются вымышленными, если не оговорено обратное.

© ООО «ЭР СИ О», 2021. Все права защищены.

ЭР СИ О, Russian Context Optimizer, RCO являются охраняемыми товарными знаками.

ООО «ЭР СИ О» может являться правообладателем патентов и заявок, поданных на получение патента, товарных знаков и объектов авторского права, которые имеют отношение к содержанию данного документа.

Предоставление вам данного документа не означает передачи какой-либо лицензии на использование данных патентов, товарных знаков и объектов авторского права, за исключением использования, явно оговоренного в лицензионном соглашении ООО «ЭР СИ О».

Все другие названия юридических лиц и изделий являются охраняемыми товарными знаками или товарными знаками, принадлежащими их владельцам.

## Оглавление

Общая структура запроса .....	7
Ограничения реализации .....	8
Реализация и использование интерфейса .....	9
ZoomClear – Удаление всех объектов Zoom .....	9
Запрос: .....	9
Ответ: .....	9
RunQuery – Проведение запроса .....	9
Запрос: .....	9
Ответ: .....	10
SelectionDel – Удаление запроса .....	10
Запрос: .....	10
Ответ: .....	10
MarkDel – Удаление объекта разметки .....	11
Запрос: .....	11
Ответ: .....	11
GetDocList – Получение части списка документов .....	11
Запрос: .....	11
Ответ: .....	12
GetDocText – Получение текста документа .....	12
Запрос: .....	12
Ответ: .....	13
GetDocPages – Получение всех страниц документа .....	13
Запрос: .....	13
Ответ: .....	13
GetDocs – Получение документов из выборки .....	14
Запрос: .....	14
Ответ: .....	14
RunStat – Проведение статистики по выборке .....	15
Запрос: .....	15
Ответ: .....	16
StatGet – Получение статистики по выборке .....	16
Запрос: .....	17
Ответ: .....	17

StatDel – Удаление статистики .....	18
Запрос:.....	18
Ответ:.....	18
SelectionGetInfo – Выборка: Получение информации.....	18
Запрос:.....	18
Ответ:.....	18
SelectionSortByField – Выборка: сортировка по полю .....	19
Запрос:.....	19
Ответ:.....	19
SelectionSortByStat – Выборка: сортировка по статистике .....	19
Запрос:.....	19
Ответ:.....	20
SelectionUnSort – Выборка: сброс сортировки .....	20
Запрос:.....	20
Ответ:.....	20
SelectionGetObjectStat – Выборка: Получение статистики объектов .....	20
Запрос:.....	20
Ответ:.....	21
SelectionGetFieldStat – Выборка: Получение статистики слов поля .....	21
Запрос:.....	21
Ответ:.....	21
StatGetDocItems – Статистика: получение статистики по документу.....	22
SelectionGetFoundWords – Выборка: получение списка найденных слов .....	22
Запрос:.....	22
Ответ:.....	22
BaseGetList – Получение списка баз .....	23
Запрос:.....	23
Ответ:.....	23
BaseGetInfo – Получение информации о базе.....	23
Запрос:.....	24
Ответ:.....	24
BaseChangeState – База: Изменение состояния .....	26
Запрос:.....	26
Ответ:.....	26
BaseCheckAlive – База: Проверка на работоспособность .....	26

BaseCheckDocAccess – База: Проверка доступности документа .....	26
BaseReopenSynonyms – База: Переоткрытие списка синонимов .....	26
BaseGetSynonymsContent – База: Получение списка синонимов .....	26
BaseSetSynonymsContent – База Загрузка списка синонимов .....	27
RegistryGetKeys – Реестр: получение ключей .....	27
Запрос: .....	27
Ответ: .....	27
RegistryGetData – Реестр: получение элементов данных .....	27
Запрос: .....	28
Ответ: .....	28
RegistryAddKey – Реестр: добавление ключа .....	28
Запрос: .....	28
Ответ: .....	29
RegistryRenameKey – Реестр: переименование ключа .....	29
Запрос: .....	29
Ответ: .....	29
RegistryDelKey – Реестр: удаление ключа .....	29
Запрос: .....	29
Ответ: .....	30
RegistryAddData – Реестр: добавление элемента данных .....	30
Запрос: .....	30
Ответ: .....	30
RegistryDelData – Реестр: удаление элемента данных .....	30
Запрос: .....	30
Ответ: .....	30
RegistrySave – Реестр: сохранение .....	31
Запрос: .....	31
Ответ: .....	31
RegistryReopen – Реестр: переоткрытие .....	31
Запрос: .....	31
Ответ: .....	31
RegistryIsChanged – Реестр: проверка изменений .....	31
Запрос: .....	31
Ответ: .....	32
LoadDoc – Загрузка документа .....	32

Запрос:.....	32
Ответ:.....	32
LoadDocCheck – Проверка загрузки документа .....	33
Запрос:.....	33
Ответ:.....	33
RemoveDocs – Удаление документов из базы.....	33
Запрос:.....	33
Ответ:.....	34
}.....	34

## Общая структура запроса

В целом запрос к серверу выглядит таким образом:

```
{  
  "Query":  
  [  
    {  
      <Операция 1>  
    },  
    {  
      <Операция 2>  
    }  
  ]  
}
```

Запросы <Операция 1> и <Операция 2> содержат один из описанных далее запросов на выполнение операций с сервером и выполняются последовательно. Эти операции называются далее «группой операций».

Ответ формируется в том же порядке:

```
{  
  "Response":  
  [  
    {  
      <Результат операции 1>  
    },  
    {  
      <Результат операции 2>  
    }  
  ]  
}
```

Запросы и ответы имеют общий формат:

```
{  
  "Operation": "<Имя операции>";  
  <Параметры запроса / результат выполнения операции>  
}
```

Каждый ответ включает атрибут "Status" со значением "Ok" – при успешном выполнении, "Error" – при возникновении ошибки, "InProgress" – в случае запуска асинхронного процесса.

Если при выполнении операции возникла ошибка, то выдается сообщение об ошибке в следующем формате:

```
{  
  "Operation": "<Имя операции>",  
  "Status": "Error",  
  "ErrorInfo":  
  {  
    "Code": <Код ошибки>,  
    "Message": "<Сообщение об ошибке>"  
  }  
}
```

Внутри ErrorInfo может присутствовать дополнительная информация об ошибке (зависит от выполняемой операции).

Если в параметрах операции предусмотрен идентификатор объекта, то в качестве него может быть передано специальное значение "#". В этом случае будет использован последний объект данного типа, использованный или созданный в рамках данного запроса.

## Ограничения реализации

На первом этапе, пока у нас не вынесена работа с объектами маркировки, в их качестве будем использовать запросы.

Идентификатор страницы - комбинация имени базы и номера страницы в базе "<Имя базы>#<Номер страницы>". Впоследствии сделаем что-то более интересное, в том числе с сохранением идентификатора при перезагрузке из базы в базу.

Идентификатор документа - для одностраничных баз совпадает с идентификатором страницы. Впоследствии будем использовать идентификатор многостраничных документов, но надо обеспечить быстрый поиск и сопоставление базам.

## Реализация и использование интерфейса

Интерфейс реализован в виде HTTP-сервиса. Сервис реализован на Python 3.7 под Flask.

Для запуска сервиса (ядро Zoom SDK должно быть предварительно установлено)

- скопировать файлы,
- переименовать файл `run.py.sample` в `run.py`,
- отредактировать его для установки параметров запуска и подключения к ядру
- установить переменную `pythonpath` в значение пути к ядру Zoom
- запустить `python run.py`.

После запуска сервис принимает POST запросы по адресу

`http://<server name>:<port>/WebAPI`

Для тестирования можно использовать форму

`http://<server name>:<port>/static/_TestAPI.htm`

## ZoomClear – Удаление всех объектов Zoom

Удаляет созданные ранее объекты выборки, маркировки, статистики.

### Запрос:

```
{  
  
  "Operation": "ZoomClear"  
  
}
```

### Ответ:

```
{  
  
  "Operation": "ZoomClear",  
  
  "Status": "Состояние"  
  
}
```

## RunQuery – Выполнение запроса

Производит запрос на поиск документов. На сервере создаются объекты выборки и разметки текста, которым присваиваются идентификаторы. После проведения объект запроса доступен для получения информации. По завершении работы с объектами их необходимо удалить (операции `DelSelection` и `DelMark`).

### Запрос:

```
{  
  
  "Operation": "RunQuery",
```

```
"BaseName": "Имя базы",
"QueryText": "Текст запроса",
"QueryParams":
{
  "Flags": "Флаги запроса",
  "ConfidenceCats": "Категории секретности",
  "SimpleWeight": "Доля веса простого запроса при наличии уточнения (число)",
  "DelPartNotFull": "Доля удаляемых неполных документов (число)",
  "MinDiffWeightNotFull": "Мин. относительная разница весов неполных документов (число)",
  "NearWeight": "Доля оценки близости искомых слов в весе документа (число)"
}
}
```

**Ответ:**

```
{
  "Operation": "RunQuery",
  "Status": "Состояние",
  "DocQuant": "Количество документов (число)",
  "SelectionId": "Идентификатор выборки",
  "MarkId": "Идентификатор объекта разметки"
}
```

**SelectionDel – Удаление запроса**

Удаляет созданный ранее объект выборки.

**Запрос:**

```
{
  "Operation": "SelectionDel",
  "SelectionId": "Идентификатор выборки"
}
```

**Ответ:**

```
{
```

```
"Operation": "SelectionDel",  
  
"Status": "Состояние"  
  
}
```

## MarkDel – Удаление объекта разметки

Удаляет созданный ранее объект разметки.

### Запрос:

```
{  
  
"Operation": "MarkDel",  
  
"MarkId": "Идентификатор объекта разметки"  
  
}
```

### Ответ:

```
{  
  
"Operation": "MarkDel",  
  
"Status": "Состояние"  
  
}
```

## GetDocList – Получение части списка документов

Получение списка входящих в выборку страниц.

В качестве DocQuant может быть задано значение «-1», тогда будут выданы все найденные документы.

Параметр PageMode может принимать следующие значения:

- "Found" – выдаются все входящие в выборку (найденные по запросу) страницы документа
- "First" – выдается только первая (в порядке сортировки) страница
- "Base64" – список страниц выдается в формате BASE64 (для будущего многобазового слияния).(?)

### Запрос:

```
{  
  
"Operation": "GetDocList",  
  
"SelectionId": "Идентификатор выборки",  
  
"DocFirst": "Номер первого документа (число)",  
  
"DocQuant": "Количество документов (число)",  
  
}
```

```
"PageMode": "Режим выдачи списка страниц"
}
```

### Ответ:

```
{
  "Operation": "GetDocList",
  "Status": "Состояние",
  "Documents":
  [
    {
      "DocNum": "Порядковый номер документа (число)",
      "DocId": "Идентификатор документа",
      "PageQuant": "Количество страниц в документе (число)",
      "Pages": ["Идентификатор страницы", "Идентификатор страницы"]
    },
  ]
}
```

### GetDocText – Получение текста документа

Формирует текст страницы документа по номеру страницы в базе. Текст формируется в соответствии с заданными параметрами. Если переданы идентификаторы объектов разметки или статистики, то текст будет соответствующим образом размечен. В качестве идентификатора может передано значение «#», тогда будет использован последний построенный в группе объект.

### Запрос:

```
{
  "Operation": "GetDocText",
  "Pages": ["Идентификатор страницы", "Идентификатор страницы"]
  "Scheme": "Схема - имя или текст",
  "Format": "Формат восстановления",
  "Flags": "Флаги",
  "FragmentLength": "Длина восстанавливаемого фрагмента (число)",
  "ContextLength": "Длина контекста восстановления (число)",
}
```

```
"MarkId": "Идентификатор объекта разметки",  
"StatId": "Идентификатор объекта статистики",  
}
```

#### Ответ:

```
{  
  "Operation": "GetDocText",  
  "Status": "Состояние",  
  "Text": ["Текст страницы", "Текст страницы"]  
}
```

### GetDocPages – Получение всех страниц документа

Возвращает список всех страниц документа.

Документ может быть идентифицирован идентификатором документа или любой из его страниц. При передаче обоих параметров приоритет имеет идентификатор документа.

#### Запрос:

```
{  
  "Operation": "GetDocPages",  
  "DocId": "Идентификатор документа",  
  "PageId": "Идентификатор страницы"  
}
```

#### Ответ:

```
{  
  "Operation": "GetDocPages",  
  "Status": "Состояние",  
  "MultiDocId": "Идентификатор мультидокумента",  
  "PageQuant": "Количество страниц в документе (число)",  
  "Pages": ["Идентификатор страницы", "Идентификатор страницы"]  
}
```

## GetDocs – Получение документов из выборки

Выдает пакетную информацию по документам выборки. Параметры соответствуют операциям GetDocList и GetDocText.

Дополнительно передаются параметры DocStatMode и DocStatQuant, определяющие выдачу слов инфопортрета. DocStatMode принимает значения «NO», «WORDS», «COMBOS», «UNITED».

### Запрос:

```
{  
  
  "Operation": "GetDocs",  
  
  "SelectionId": "Идентификатор выборки",  
  
  "DocFirst": "Номер первого документа (число)",  
  
  "DocQuant": "Количество документов (число)",  
  
  "PageMode": "Режим выдачи списка страниц"  
  
  "Schemes": ["Схема 1 - имя или текст", "Схема 2 - имя или текст"]  
  
  "Format": "Формат восстановления",  
  
  "Flags": "Флаги",  
  
  "FragmentLength": "Длина восстанавливаемого фрагмента",  
  
  "MarkId": "Идентификатор объекта разметки",  
  
  "StatId": "Идентификатор объекта статистики",  
  
  "ContextLength": "Длина контекста восстановления",  
  
  "DocStatMode": "Режим вывода статистики по документу WORD|PAIR|COMBO",  
  
  "DocStatQuant": "Количество элементов статистики по документу",  
  
  "DocObjMode": "Режим вывода объектов по документу",  
  
  "DocObjFilter": "Фильтр вывода объектов по документу",  
  
  "DocObjQuant": "Количество объектов по документу"  
  
}
```

### Ответ:

```
{  
  
  "Operation": "GetDocs",  
  
  "Status": "Состояние",  
  
  "Documents":
```

```
[
  {
    "PageQuant": "Количество страниц в документе",
    "Pages":
    [
      {
        "Number": "номер страницы",
        "PageId": "Идентификатор страницы"
        "Text": ["Текст документа по схеме 1", "Текст документа по схеме 2"],
        "ArchiveType": "Тип архивного документа"
      }
    ]
    "Statistics": [{"Item": "Текст элемента", "Weight": "Вес"}, ],
    "Objects": ["Объект", ],
  },
  "Weight": "Вес документа в текущей сортировке"
]
```

## RunStat – Проведение статистики по выборке

Для объекта выборки строит объект статистики, которому присваивается идентификатор. После проведения объект статистики доступен для получения информации. По завершении работы с объектом его необходимо удалить (операция DelStat).

В качестве StatParams может быть передан null, тогда статистика будет проведена с параметрами по-умолчанию.

### Запрос:

```
{
  "Operation": "RunStat",
  "SelectionId": "Идентификатор выборки",
  "FieldName": "Имя поля для проведения статистики",
```

```
"Flags": "Флаги статистики",  
"StatParams":  
{  
  "lowBase": "Минимальная энтропия по базе (число)",  
  "upBase": "Максимальная энтропия по базе (число)",  
  "lowSelect": "Минимальная энтропия по выборке (число)",  
  "upSelect": "Максимальная энтропия по выборке (число)",  
  "lowBaseRel": "Относительная минимальная энтропия по базе (число)",  
  "upBaseRel": "Относительная максимальная энтропия по базе (число)",  
  "lowSelectRel": "Относительная минимальная энтропия по выборке (число)",  
  "upSelectRel": "Относительная максимальная энтропия по выборке (число)",  
  "ProcesDocQnt": "Количество обрабатываемых документов (число)",  
  "StatCalcMode": "Способ расчета статистики (число)",  
  "StatUnitedWeight": "Весовой коэфф. словосочетаний для объедин. статистики (число)"  
}  
}
```

### Ответ:

```
{  
  "Operation": "RunStat",  
  "Status": "Состояние",  
  "StatId": "Идентификатор объекта статистики"  
}
```

## StatGet – Получение статистики по выборке

Выдает размер, качество и список элементов инфопортрета. Расстояние между словами (опционально).

Параметр Flags определяет состав выдаваемой информации:

- 0x01 - выдавать элементы
- 0x02 - выдавать веса элементов
- 0x04 - выдавать расстояния
- 0x08 - выдавать расстояния от предыдущих элементов

**Запрос:**

```
{  
  
  "Operation": "StatGet",  
  
  "StatId": "Идентификатор объекта статистики",  
  
  "Flags": "Флаги выдачи статистики",  
  
  "ItemType": "Тип элементов статистики - WORD|PAIR|COMBO",  
  
  "ItemFirst": "Номер первого элемента (число)",  
  
  "ItemQuant": "Количество элементов (число)"  
  
}
```

**Ответ:**

```
{  
  
  "Operation": " StatGet ",  
  
  "Status": "Состояние",  
  
  "Items":  
  
  [  
  
    {  
  
      "Index": "Порядковый номер (число)",  
  
      "Item": "Текст объекта",  
  
      "Weight": "Вес объекта (число)"  
  
    },  
  
  ]  
  
  "Distances":  
  
  [  
  
    {  
  
      "ItemFrom": "Порядковый номер первого элемента (число)",  
  
      "ItemTo": "Порядковый номер второго элемента (число)",  
  
      "Distance": "Расстояние (число)"  
  
    },  
  
  ]  
  
}
```

```
}
```

## StatDel – Удаление статистики

Удаляет созданный ранее объект статистики.

### Запрос:

```
{  
  
  "Operation": "StatDel",  
  
  "StatId": "Идентификатор объекта статистики"  
  
}
```

### Ответ:

```
{  
  
  "Operation": "StatDel",  
  
  "Status": "Состояние"  
  
}
```

## SelectionGetInfo – Выборка: Получение информации

Выдает информацию о созданном ранее объекте выборки.

### Запрос:

```
{  
  
  "Operation": "SelectionGetInfo",  
  
  "SelectionId": "Идентификатор выборки"  
  
}
```

### Ответ:

```
{  
  
  "Operation": "SelectionGetInfo",  
  
  "Status": "Состояние",  
  
  "BaseName": "Имя базы",  
  
  "QueryText": "Текст запроса",  
  
  "Flags": "Флаги запроса",  
  
  "DocQuant": "Количество документов (число)",  
  
  "PartsInfo": "Информация о частях запроса"
```

```
}
```

## SelectionSortByField – Выборка: сортировка по полю

Сортирует документы выборки по полю.

### Запрос:

```
{  
  "Operation": "SelectionSortByField",  
  "SelectionId": "Идентификатор выборки",  
  "FieldName": "Имя поля для сортировки",  
  "Order": "Направление сортировки Asc|Desc"  
}
```

### Ответ:

```
{  
  "Operation": "SelectionSortByField",  
  "Status": "Состояние"  
}
```

## SelectionSortByStat – Выборка: сортировка по статистике

Сортирует документы выборки по набору инфопортретов.

### Запрос:

```
{  
  "Operation": "SelectionSortByStat",  
  "SelectionId": "Идентификатор выборки",  
  "StatObjects":  
  [  
    {  
      "StatId": "Идентификатор объекта статистики",  
      "Weight": "Вес объекта (число)"  
    },  
  ],  
}
```

```
"Order": "Направление сортировки Asc|Desc"  
}
```

**Ответ:**

```
{  
  
  "Operation": "SelectionSortByStat",  
  
  "Status": "Состояние"  
}
```

**SelectionUnSort – Выборка: сброс сортировки**

Сбрасывает произведенную ранее сортировку.

**Запрос:**

```
{  
  
  "Operation": "SelectionUnSort",  
  
  "SelectionId": "Идентификатор выборки",  
  
  "Order": "Направление сортировки Asc|Desc"  
}
```

**Ответ:**

```
{  
  
  "Operation": "SelectionUnSort",  
  
  "Status": "Состояние"  
}
```

**SelectionGetObjectStat – Выборка: Получение статистики объектов****Запрос:**

```
{  
  
  "Operation": "SelectionGetObjectStat",  
  
  "SelectionId": "Идентификатор выборки",  
  
  "ObjFirst": "Номер первого объекта (число)",  
  
  "ObjQuant": "Количество объектов (число)",  
  
  "Flags": "Флаги выдачи списка",  
  
  "Filter": "Фильтр вывода объектов"
```

```
}
```

**Ответ:**

```
{  
  
  "Operation": "SelectionGetObjectStat",  
  
  "Status": "Состояние",  
  
  "Objects":  
  
  [  
  
    {  
  
      "ObjNum": "Порядковый номер объекта (число)",  
  
      "Attribute": "Атрибут объекта",  
  
      "EntryQuant": "Количество вхождений (число)"  
  
    },  
  
  ]  
  
}
```

**SelectionGetFieldStat – Выборка: Получение статистики слов поля****Запрос:**

```
{  
  
  "Operation": "SelectionGetFieldStat",  
  
  "SelectionId": "Идентификатор выборки",  
  
  "WordFirst": "Номер первого слова (число)",  
  
  "WordQuant": "Количество слов (число)",  
  
  "Flags": "Флаги выдачи списка"  
  
}
```

**Ответ:**

```
{  
  
  "Operation": "SelectionGetFieldStat",  
  
  "Status": "Состояние",  
  
  "Words":
```

```
[  
  {  
    "WordNum": "Порядковый номер слова (число)",  
    "Word": "Слово",  
    "EntryQuant": "Количество вхождений (число)"  
  },  
]  
}
```

### **StatGetDocItems – Статистика: получение статистики по документу**

Формат запроса и ответа аналогичен операции GetStat, но в параметрах запроса дополнительно указывается идентификатор страницы.

### **SelectionGetFoundWords – Выборка: получение списка найденных слов**

#### **Запрос:**

```
{  
  "Operation": "SelectionGetFoundWords ",  
  "SelectionId": "Идентификатор выборки",  
  "WordQuant": "Количество слов (число)"  
}
```

#### **Ответ:**

```
{  
  "Operation": "SelectionGetFoundWords ",  
  "Status": "Состояние",  
  "Words": ["Слово", ]  
}
```

### **BaseGetList – Получение списка баз**

#### **Запрос:**

```
{
```

```
"Operation": "BaseGetList"
}
```

### Ответ:

```
{
  "Operation": " BaseGetList ",
  "Status": "Состояние"
  "BaseQuant": "Количество доступных баз (число)",
  "Bases":
  [
    {
      "BaseName": "Имя базы",
      "BaseType": "Тип базы - LOCAL|FAMILY|REMOTE",
      "BaseState": "Состояние базы -OPENED|CLOSED"
    },
  ]
}
```

## BaseGetInfo – Получение информации о базе

Версия, UID, структура, поля, размеры, даты ...

Формат незавершенный. Потом будем дополнять.

Части выдаются при указании соответствующих флагов (Flags):

- 0x01 - выдавать информацию о полях
- 0x02 - выдавать информацию о версии
- 0x04 - выдавать информацию о файлах
- 0x08 - выдавать информацию о стеках

### Запрос:

```
{
  "Operation": "BaseGetInfo",
  "BaseName": "Имя базы",
```

```

    "Flags": "Флаги выдачи информации"
}

```

### Ответ:

```

{
# Основная часть - выдается всегда

    "Operation": "BaseGetInfo",
    "Status": "Состояние",
    "BaseId": "Идентификатор базы",
    "BaseType": "Тип базы - LOCAL|FAMILY|REMOTE",
    "BaseState": "Состояние базы -OPENED|CLOSED",
    "QuantDoc": "Количество документов, включая удаленные (число)",
    "QuantDelDoc": "Количество удаленных документов (число)",
    "BaseFlags": "Флаги базы @see EnpBaseFlags (число)",
    "QuantWords": "Количество слов в базе по всем полям (число)",
# Информация о полях

    "QuantFields": "Количество полей в базе без учета служебных (число)",
    "Fields":
    [
        {
            "Type": "Тип поля @see EnpBaseFieldTypeCode (число)",
            "Precision": "Точность представления чисел с фиксированной точностью. Для остальных типов - -1. (число)",
            "Flags": "Флаги поля @see EnpBaseFieldFlags (число)",
            "FldDicSize": "Количество слов в полевом словаре (число)",
        },
    ]
# Информация о версии

    "BaseContent": "Флаги состава базы @see EnpBaseContentFlags (число)",
    "MainFileVersion": "Версия файлов основной базы (число)",

```

```
"ArcFileVersion": "Версия файлов архива, или 0 если архива нет (число)",
```

```
"MorfVersion": "Версия морфологии",
```

```
# Информация о файлах
```

```
"Size": "Размер файлов (число)",
```

```
"Date": "Дата обновления файлов",
```

```
"ShortName": "Короткое имя базы",
```

```
"Path": "Путь к файлам базы",
```

```
"QuantTrans": "Количество транзакций (число)",
```

```
# Информация о стеках
```

```
"Stacks":
```

```
[
```

```
{
```

```
  "Name": "Название стека",
```

```
  "ItemQuant": "Количество элементов в стеке",
```

```
  "Items":
```

```
  [
```

```
    {
```

```
      "Number": "Номер стека (число)",
```

```
      "Path": "Путь к файлам экземпляра",
```

```
      "QuantDoc": "Количество документов, включая удаленные (число)",
```

```
      "QuantDelDoc": "Количество удаленных документов (число)",
```

```
      "QuantWords": "Количество слов в базе по всем полям (число)",
```

```
      "Size": "Размер файлов (число)",
```

```
      "Date": "Дата обновления файлов",
```

```
      "QuantTrans": "Количество транзакций (число)"
```

```
    }
```

```
  ]
```

```
}
```

```
]
}
```

### **BaseChangeState – База: Изменение состояния**

Выполняет операцию, возможно, приводящую к изменению состояния базы. Если состояние при выполнении операций OPEN|CLOSE не должно меняться, то ничего не делается.

#### **Запрос:**

```
{
  "Operation": "BaseChangeState",
  "BaseName": "Имя базы",
  "Action": "Действие над базой - OPEN|CLOSE|REOPEN"
}
```

#### **Ответ:**

```
{
  "Operation": "BaseSetState",
  "Status": "Состояние"
}
```

### **BaseCheckAlive – База: Проверка на работоспособность**

### **BaseCheckDocAccess – База: Проверка доступности документа**

### **BaseReopenSynonyms – База: Переоткрытие списка синонимов**

### **BaseGetSynonymsContent – База: Получение списка синонимов**

### **BaseSetSynonymsContent – База Загрузка списка синонимов**

## RegistryGetKeys – Реестр: получение ключей

В качестве глубины (Depth) может быть передано (-1), тогда выдаются все ключи.

Флаги (Flags):

- 0x01 - выдавать в виде дерева
- 0x02 - выдавать полные имена

### Запрос:

```
{  
  
  "Operation": "RegistryGetKeys",  
  
  "Parent": "Путь к родителю, при пустом - от корня",  
  
  "Depth": "Количество слов уровней (число)",  
  
  "Flags": "Флаги выдачи списка (число) "  
  
}
```

### Ответ:

```
{  
  
  "Operation": "RegistryGetKeys",  
  
  "Status": "Состояние",  
  
  "Keys": {"Ключ": {"Ключ": {}}, }  
  
}
```

Или

```
{  
  
  "Operation": "RegistryGetKeys",  
  
  "Status": "Состояние",  
  
  "Keys": ["Ключ", "Ключ", ]  
  
}
```

## RegistryGetData – Реестр: получение элементов данных

Флаги (Flags):

- 0x01 - выдавать в натуральном виде (без расширения строк)
- 0x02 - использовать регулярное выражение для поиска

Типы элементов данных (они же используются в RegistryAddData):

- 'b' - логический, значения 0/1

- 'i' - целое число
- 'f' - число с плавающей точкой
- 's' - строка

Формат передачи значения определяется типом данных.

### Запрос:

```
{  
  
  "Operation": "RegistryGetData",  
  
  "Parent": "Путь к родителю ",  
  
  "Flags": "Флаги выдачи списка (число) "  
  
  "Name": "Имя или маска имени (регулярное выражение), при пустом выдаются все"  
  
}
```

### Ответ:

```
{  
  
  "Operation": "RegistryGetData",  
  
  "Status": "Состояние",  
  
  "Data":  
  
  [  
  
    {  
  
      "Name": "Имя элемента данных",  
  
      "Type": "Тип элемента данных",  
  
      "Value": "Значение элемента данных"  
  
    },  
  
  ]  
  
}
```

## RegistryAddKey – Реестр: добавление ключа

### Запрос:

```
{  
  
  "Operation": "RegistryAddKey",  
  
  "Parent": "Путь к родителю ",  
  
  "Name": "Имя ключа"
```

```
}
```

**Ответ:**

```
{  
  
  "Operation": "RegistryAddKey",  
  
  "Status": "Состояние"  
  
}
```

**RegistryRenameKey – Реестр: переименование ключа****Запрос:**

```
{  
  
  "Operation": "RegistryRenameKey",  
  
  "OldName": "Старое имя ключа (полностью от корня)",  
  
  "NewName": "Новое имя ключа (только имя)"  
  
}
```

**Ответ:**

```
{  
  
  "Operation": "RegistryRenameKey",  
  
  "Status": "Состояние"  
  
}
```

**RegistryDelKey – Реестр: удаление ключа****Запрос:**

```
{  
  
  "Operation": "RegistryDelKey",  
  
  "Path": "Путь к ключу"  
  
}
```

**Ответ:**

```
{  
  
  "Operation": "RegistryDelKey",  
  
  "Status": "Состояние"  
  
}
```

```
}
```

## RegistryAddData – Реестр: добавление элемента данных

### Запрос:

```
{  
  "Operation": "RegistryAddData",  
  "Parent": "Путь к родителю ",  
  "Name": "Имя элемента данных",  
  "Type": "Тип элемента данных",  
  "Value": "Значение элемента данных"  
}
```

### Ответ:

```
{  
  "Operation": "RegistryAddData",  
  "Status": "Состояние"  
}
```

## RegistryDelData – Реестр: удаление элемента данных

### Запрос:

```
{  
  "Operation": "RegistryDelData",  
  "Parent": "Путь к родителю ",  
  "Name": "Имя элемента данных"  
}
```

### Ответ:

```
{  
  "Operation": "RegistryDelData",  
  "Status": "Состояние"  
}
```

## RegistrySave – Реестр: сохранение

### Запрос:

```
{  
  
  "Operation": "RegistrySave"  
  
}
```

### Ответ:

```
{  
  
  "Operation": "RegistrySave",  
  
  "Status": "Состояние"  
  
}
```

## RegistryReopen – Реестр: переоткрытие

### Запрос:

```
{  
  
  "Operation": "RegistryReopen"  
  
}
```

### Ответ:

```
{  
  
  "Operation": "RegistryReopen",  
  
  "Status": "Состояние"  
  
}
```

## RegistryIsChanged – Реестр: проверка изменений

### Запрос:

```
{  
  
  "Operation": "RegistryIsChanged"  
  
}
```

### Ответ:

```
{  
  
  "Operation": "RegistryIsChanged",  
  
}
```

```
"IsChanged": "0|1 (число)",  
  
"Status": "Состояние"  
  
}
```

## LoadDoc – Загрузка документа

Отправляет документ на загрузку. Формирует файл и помещает во входную директорию базы. Имя файла строится по идентификатору

### Запрос:

```
{  
  
  "Operation": "LoadDoc",  
  
  "BaseName": "Имя базы",  
  
  "LoadItem": "Имя элемента цикла загрузки",  
  
  "DocId": "Идентификатор документа",  
  
  "Fields":  
  
  [  
  
    {  
  
      "Name": "Имя поля",  
  
      "Value": "Значение поля"  
  
    },  
  
  ]  
  
}
```

### Ответ:

```
{  
  
  "Operation": "LoadDoc",  
  
  "Status": "Состояние"  
  
}
```

## LoadDocCheck – Проверка загрузки документа

### Запрос:

```
{  
  
  "Operation": "LoadDocCheck",
```

```
"BaseName": "Имя базы",  
"LoadItem": "Имя элемента цикла загрузки",  
"Docs":["Идентификатор документа"],  
}
```

**Ответ:**

```
{  
  "Operation": "LoadDocCheck",  
  "Status": "Состояние",  
  "Docs": [  
    {"DocId": "Идентификатор документа",  
      "LoadState": "Состояние загрузки - LOADED|WAIT|ERROR",  
      "AddInfo": "Дополнительная информация - при ошибке"},  
  ],  
}
```

**RemoveDocs – Удаление документов из базы****Запрос:**

```
{  
  "Operation": "RemoveDocs",  
  "BaseName": "Имя базы",  
  "QueryText": "Текст запроса",  
  "Flags": "Флаги выдачи списка (число)"  
}
```

**Ответ:**

```
{  
  "Operation": "LoadDocCheck",  
  "Status": "Состояние",  
  "Deleted": "Количество удалённых документов"
```

}