

Документация по API Обновление номенклатуры и загрузка изображений

Оглавление

##Скрипт для получения типов цен	2
### Описание	2
### Входные данные	2
### Выходные данные	2
#### Пример ошибки	3
### Примечание по ID:	3
### Примечание по URL	3
##Скрипт для обновления цен продуктов	3
### Описание	4
### Входные данные	4
### Пример 1: Один товар с несколькими ценами	4
### Пример 2: Несколько товаров с одной ценой каждый	5
### Пример 3: Несколько товаров с несколькими ценами каждый	5
### Выходные данные	6
##Скрипт для получения складов	7
### Описание	7
### Входные данные	7
### Выходные данные	7
### Примечание по ID:	8
## Скрипт для обновления количества товаров на складах	8
### Описание	8
### Входные данные	8
### Пример 1: Один товар с несколькими количествами на разных складах	9
### Пример 2: Несколько товаров с разным количеством на одном складе	9
### Пример 3: Несколько товаров с несколькими количествами на разных складах	10
### Пример 4: Несколько товаров с разными количествами на одном складе	11
### Выходные данные	11
Примеры работы со скриптом PRODUCTS	12
## Скрипт для получения свойств инфоблока и свойств элемента	14
### Описание	14
### Входные данные	14
### Выходные данные	14
### Пример использования	15
#### Запрос на получение свойств инфоблока	15
#### Запрос на получение свойств элемента инфоблока	16

### Обработка ошибок	16
## Скрипт для поиска раздела и его создания	17
### Описание	17
### Входные данные	17
### Выходные данные	17
### Пример использования	18
##### Запрос на создание разделов	18
##### Ответ	19
##### Запрос с ошибкой в данных	19
##### Ответ	19
### Обработка ошибок	19
##### Примеры сообщений об ошибках:	19
##Скрипт для обновления изображений	20
### Описание	20
### Входные данные	20
### Пример 1: Один товар с одним изображением	20
### Пример 2: Несколько товаров с несколькими изображением каждый	21
### Выходные данные	21

##Скрипт для получения типов цен

Описание

Этот скрипт принимает JSON-данные через HTTP-запрос на адрес <https://eco-balt.ru/api/pricetypes/> , проверяет токен безопасности и возвращает список типов цен из модуля каталога Bitrix24.

Входные данные

Скрипт ожидает JSON-данные в следующем формате:

```
```json
{
 "token": "<ваш токен>"
}
```
```

Выходные данные

Скрипт возвращает JSON-ответ с результатом выполнения и списком типов цен:

```
```json
{
 "result": "success" | "error",
 "errmsg": ["Error message 1", "Error message 2"],
 "data": [
 {
 "ID": "1",
 "NAME": "BASE"
 },
 {
 "ID": "2",
 "NAME": "RETAIL"
 }
]
}
```

#### #### Пример ошибки

```
```json
{
  "result": "error",
  "errmsg": ["Ошибка декодирования JSON: Syntax error"]
}
...`
```

Примечание по ID:

Розничная цена на сайте - ID - 1
Базовая цена на сайте - ID - 160
Оптовые цены на сайте - ID - 159

Примечание по URL

Убедитесь, что URL запроса содержит слэш в конце (`/api/pricetypes/`), чтобы избежать перенаправлений и ошибок.

##Скрипт для обновления цен продуктов

Описание

Этот скрипт принимает JSON-данные через HTTP-запрос на адрес </api/products/>, проверяет токен безопасности, находит элементы в каталоге Bitrix по их GUID в 1С (Внешний код в 1С) и обновляет или добавляет цены для этих элементов.

Входные данные

Скрипт ожидает JSON-данные в следующем формате:

```
```json
{
 "token": "5f)m*EDs",
 "products": [
 {
 "XML_ID": "some-xml-id",
 "prices": [
 {
 "id": 1,
 "price": 100.0,
 "currency": "RUB"
 }
]
 }
]
}
```

Скрипт поддерживает отправку нескольких товаров и/или нескольких цен

### ### Пример 1: Один товар с несколькими ценами

```
```json
{
  "token": "5f)m*EDs",
  "products": [
    {
      "XML_ID": "product-xml-id-1",
      "prices": [
        {
          "id": 1,
          "price": 100.0,
          "currency": "RUB"
        },

```

```

    {
      "id": 2,
      "price": 150.0,
      "currency": "USD"
    }
  ]
}
]
}
...

```

Пример 2: Несколько товаров с одной ценой каждый

```

```json
{
 "token": "5f)m*EDs",
 "products": [
 {
 "XML_ID": "product-xml-id-1",
 "prices": [
 {
 "id": 1,
 "price": 100.0,
 "currency": "RUB"
 }
]
 },
 {
 "XML_ID": "product-xml-id-2",
 "prices": [
 {
 "id": 1,
 "price": 200.0,
 "currency": "RUB"
 }
]
 }
]
}
...

```

### ### Пример 3: Несколько товаров с несколькими ценами каждый

```

```json

```

```

{
  "token": "5f)m*EDs",
  "products": [
    {
      "XML_ID": "product-xml-id-1",
      "prices": [
        {
          "id": 1,
          "price": 100.0,
          "currency": "RUB"
        },
        {
          "id": 2,
          "price": 150.0,
          "currency": "USD"
        }
      ]
    },
    {
      "XML_ID": "product-xml-id-2",
      "prices": [
        {
          "id": 1,
          "price": 200.0,
          "currency": "RUB"
        },
        {
          "id": 2,
          "price": 250.0,
          "currency": "USD"
        }
      ]
    }
  ]
}

```

Выходные данные

Скрипт возвращает JSON-ответ с результатом выполнения:

```

```json
{
 "result": "success" | "error",
 "errmsg": ["Error message 1", "Error message 2"],
 "output": "Detailed output messages"
}

```

...

## ##Скрипт для получения складов

### ### Описание

Этот PHP-скрипт принимает JSON-данные через HTTP-запрос на адрес `/api/storetypes/`, проверяет их на корректность и наличие токена безопасности, а затем возвращает список складов, их ID и Название в формате JSON.

### ### Входные данные

Скрипт ожидает JSON-данные в следующем формате:

```
```json
{
  "token": "5f)m*EDs"
}
```
```

### ### Выходные данные

Скрипт возвращает JSON-ответ в следующем формате:

- Если данные корректны и токен безопасности совпадает:

```
```json
{
  "result": "success",
  "errmsg": [],
  "data": [
    {
      "ID": "1",
      "TITLE": "Склад №2"
    },
    {
      "ID": "2",
      "TITLE": "Магазин"
    },
    // Другие типы складов...
  ]
}
```

...

- Если данные некорректны или токен безопасности не совпадает:

```
```json
{
 "result": "error",
 "errmsg": ["Invalid token or JSON structure"]
}
```
```

Примечание по ID:

Склад №2 - ID - 1

Магазин - ID - 2

Скрипт для обновления количества товаров на складах

Описание

Этот PHP-скрипт принимает JSON-данные через HTTP-запрос на адрес `/api/products/`, проверяет их на корректность и наличие токена безопасности, а затем обновляет количество товаров на складах в системе 1С-Битрикс. Скрипт использует функции для поиска элемента в инфоблоке и изменения количества товара на складе.

Входные данные

Скрипт ожидает JSON-данные в следующем формате:

```
```json
{
 "token": "5fm*EDs", // Токен безопасности для аутентификации запроса
 "products": [// Массив продуктов для обновления
 {
 "XML_ID": "0e2bca2a-3029-4efb-9635-2f06f8cfa99d", // Уникальный идентификатор продукта в формате 1C GUID
 "quantity": [// Массив количеств товара на различных складах
 {
 "id": 1, // Идентификатор склада
 "amount": 0 // Количество товара на складе
 }
]
 }
]
}
```
```

```

    }
  ]
}
]
}
...

```

Скрипт поддерживает отправку нескольких товаров и/или нескольких количеств на разных складах.

Пример 1: Один товар с несколькими количествами на разных складах

```

```json
{
 "token": "5f)m*EDs",
 "products": [
 {
 "XML_ID": "product-xml-id-1",
 "quantity": [
 {
 "id": 1,
 "amount": 100
 },
 {
 "id": 2,
 "amount": 150
 }
]
 }
]
}
...

```

### ### Пример 2: Несколько товаров с разным количеством на одном складе

```

```json
{
  "token": "5f)m*EDs",
  "products": [
    {
      "XML_ID": "product-xml-id-1",
      "quantity": [
        {
          "id": 1,
          "amount": 100
        }
      ]
    }
  ]
}
...

```

```

    ]
  },
  {
    "XML_ID": "product-xml-id-2",
    "quantity": [
      {
        "id": 1,
        "amount": 200
      }
    ]
  }
]
}
...

```

Пример 3: Несколько товаров с несколькими количествами на разных складах

```

```json
{
 "token": "5f)m*EDs",
 "products": [
 {
 "XML_ID": "product-xml-id-1",
 "quantity": [
 {
 "id": 1,
 "amount": 100
 },
 {
 "id": 2,
 "amount": 150
 }
]
 },
 {
 "XML_ID": "product-xml-id-2",
 "quantity": [
 {
 "id": 1,
 "amount": 200
 },
 {
 "id": 2,
 "amount": 250
 }
]
 }
]
}

```

```

]
 }
]
}
...

```

### ### Пример 4: Несколько товаров с разными количествами на одном складе

```

```json
{
  "token": "5f)m*EDs",
  "products": [
    {
      "XML_ID": "product-xml-id-1",
      "quantity": [
        {
          "id": 1,
          "amount": 100
        }
      ]
    },
    {
      "XML_ID": "product-xml-id-2",
      "quantity": [
        {
          "id": 1,
          "amount": 200
        }
      ]
    }
  ]
}

```

Выходные данные

Скрипт возвращает JSON-ответ в следующем формате:

- Если данные корректны и токен безопасности совпадает:

```

```json
{
 "result": "success",

```

```
"errmsg": [],
"output": ""
}
...
```

- Если данные некорректны или токен безопасности не совпадает:

```
```json  
{  
  "result": "error",  
  "errmsg": ["Invalid token or JSON structure"]  
}  
...
```

- Если произошли ошибки при обработке данных:

```
```json  
{
 "result": "error",
 "errmsg": [
 "ERROR: no GUID",
 "ERROR: Element with XML_ID 0e2bca2a-3029-4efb-9635-2f06f8cfa99d not found",
 "ERROR: update failed for XML_ID 0e2bca2a-3029-4efb-9635-2f06f8cfa99d, Store_ID 1"
],
 "output": ""
}
...
```

## Примеры работы со скриптом PRODUCTS

Вот пример JSON, который включает в себя как обновление цен, так и изменение количества товаров на складах для нескольких продуктов:

```
```json  
{  
  "token": "5f)m*EDs", // Токен безопасности для аутентификации запроса  
  "products": [ // Массив продуктов для обновления  
    {  
      "XML_ID": "product-xml-id-1", // Уникальный идентификатор продукта в формате 1C GUID  
      "prices": [ // Массив цен для продукта  
        {  
          "id": 1, // Идентификатор группы цен  
          "price": 100.0, // Значение цены  
          "currency": "RUB" // Валюта цены  
        },  
        {  
          "id": 2, // Идентификатор другой группы цен
```

```

        "price": 150.0, // Значение цены
        "currency": "USD" // Валюта цены
    }
],
"quantity": [ // Массив количеств товара на различных складах
    {
        "id": 1, // Идентификатор склада
        "amount": 100 // Количество товара на складе
    },
    {
        "id": 2, // Идентификатор другого склада
        "amount": 150 // Количество товара на складе
    }
]
},
{
    "XML_ID": "product-xml-id-2", // Уникальный идентификатор другого продукта в формате 1C
GUID
    "prices": [ // Массив цен для другого продукта
        {
            "id": 1, // Идентификатор группы цен
            "price": 200.0, // Значение цены
            "currency": "RUB" // Валюта цены
        },
        {
            "id": 2, // Идентификатор другой группы цен
            "price": 250.0, // Значение цены
            "currency": "USD" // Валюта цены
        }
    ],
    "quantity": [ // Массив количеств товара на различных складах
        {
            "id": 1, // Идентификатор склада
            "amount": 200 // Количество товара на складе
        },
        {
            "id": 2, // Идентификатор другого склада
            "amount": 250 // Количество товара на складе
        }
    ]
}
]
}
...

```

Скрипт для получения свойств инфоблока и свойств элемента

Описание

Этот PHP-скрипт принимает JSON-данные через HTTP-запрос на адрес `/api/properties/`, проверяет их на корректность и наличие токена безопасности, а затем возвращает список свойств инфоблока или свойства конкретного элемента инфоблока в формате JSON.

Входные данные

Скрипт ожидает JSON-данные в следующем формате:

```
``json
{
  "token": "5f)m*EDs",
  "XML_ID": "some_xml_id" // Параметр необходимый для получения свойства элемента
}
...`
```

Выходные данные

Скрипт возвращает JSON-ответ в следующем формате:

- Если данные корректны и токен безопасности совпадает:
- Если передан параметр ``XML_ID``, возвращаются свойства элемента инфоблока:

```
``json
{
  "result": "success",
  "errmsg": [],
  "data": [
    {
      "ID": "1",
      "NAME": "Property Name",
      "VALUE": "Property Value"
    },
    // Другие свойства элемента...
  ]
}
```

...

- Если параметр `XML\_ID` не передан, возвращаются свойства инфоблока:

```
```json
{
 "result": "success",
 "errmsg": [],
 "data": [
 "1 - Property Name - XML_ID",
 // Другие свойства инфоблока...
]
}
```
```

- Если данные некорректны или токен безопасности не совпадает:

```
```json
{
 "result": "error",
 "errmsg": ["Invalid token or JSON structure"]
}
```
```

Пример использования

Запрос на получение свойств инфоблока

```
```json
{
 "token": "5f)m*EDs"
}
```
```

Ответ

```
```json
{
 "result": "success",
 "errmsg": [],
 "data": [
 "1 - Property Name - XML_ID",
 // Другие свойства инфоблока...
]
}
```

```
}
...
```

#### #### Запрос на получение свойств элемента инфоблока

```
```json  
{  
  "token": "5f)m*EDs",  
  "XML_ID": "some_xml_id"  
}  
...
```

Ответ

```
```json  
{
 "result": "success",
 "errmsg": [],
 "data": [
 {
 "ID": "1",
 "NAME": "Property Name",
 "VALUE": "Property Value"
 },
 // Другие свойства элемента...
]
}
...
```

#### ### Обработка ошибок

Если JSON-данные некорректны или токен безопасности не совпадает, скрипт возвращает следующий ответ:

```
```json  
{  
  "result": "error",  
  "errmsg": ["Invalid token or JSON structure"]  
}  
...
```

Скрипт для поиска раздела и его создания

Описание

Этот PHP-скрипт принимает JSON-данные через HTTP-запрос на адрес `/api/sections/`, проверяет их на корректность и наличие токена безопасности, а затем возвращает список свойств инфоблока или свойства конкретного элемента инфоблока в формате JSON.

Входные данные

Скрипт ожидает JSON-данные в следующем формате:

```
{
  "token": "5f)m*EDs",
  "sections": [
    {
      "XML_ID": "some_xml_id",
      "Name": "Section Name",
      "Parent_XML_ID": "parent_xml_id"
    },
    // Другие разделы...
  ]
}
```

token: Токен безопасности для проверки подлинности запроса.

sections: Массив разделов инфоблока, каждый из которых содержит:

XML_ID: Уникальный идентификатор раздела.

Name: Название раздела.

Parent_XML_ID: Уникальный идентификатор родительского раздела.

Выходные данные

Скрипт возвращает JSON-ответ в следующем формате:

Если токен безопасности корректен и данные обработаны успешно:

json

```
{
  "result": "успех",
  "errmsg": [],
  "output": "Содержимое буфера вывода (если есть)"
}
```

```
}
```

Если возникли ошибки в процессе обработки (например, некорректные данные или проблемы с созданием разделов):

```
{  
  "result": "ошибка",  
  "errmsg": [  
    "ОШИБКА: Сообщение об ошибке 1",  
    "ОШИБКА: Сообщение об ошибке 2"  
  ],  
  "output": "Содержимое буфера вывода (если есть)"  
}
```

Если токен безопасности некорректен:

```
json  
  
{  
  "result": "ошибка",  
  "errmsg": ["Invalid token or JSON structure"]  
}
```

Пример использования

Запрос на создание разделов

```
{  
  "token": "5f)m*EDs",  
  "sections": [  
    {  
      "XML_ID": "section_1",  
      "Name": "Раздел 1",  
      "Parent_XML_ID": "parent_1"  
    },  
    {  
      "XML_ID": "section_2",  
      "Name": "Раздел 2",  
      "Parent_XML_ID": "parent_2"  
    }  
  ]  
}
```

```
]
}
```

Ответ

```
{
  "result": "успех",
  "errmsg": [],
  "output": "Содержимое буфера вывода (если есть)"
}
```

Запрос с ошибкой в данных

```
{
  "token": "5f)m*EDs",
  "sections": [
    {
      "Name": "Раздел без XML_ID"
    }
  ]
}
```

Ответ

```
{
  "result": "ошибка",
  "errmsg": [
    "ОШИБКА: нет XML_ID"
  ],
  "output": "Содержимое буфера вывода (если есть)"
}
```

Обработка ошибок

Если данные некорректны или токен безопасности не совпадает, скрипт возвращает соответствующее сообщение об ошибке. Возможные ошибки включают, но не ограничиваются:

- Отсутствие токена или некорректный токен.
- Отсутствие обязательного поля XML_ID.
- Ошибки при создании новых разделов.

Примеры сообщений об ошибках:

```
{
  "result": "error",
  "errmsg": ["Invalid token or JSON structure"]
}
```

##Скрипт для обновления изображений

Описание

Этот скрипт принимает JSON-данные через HTTP-запрос на адрес /api/products/, проверяет токен безопасности, находит элементы в каталоге Bitrix по их GUID в 1С (Внешний код в 1С) и обновляет или добавляет цены для этих элементов.

Входные данные

Скрипт ожидает JSON-данные в следующем формате:

```
```json
{
 "token": "5f)m*EDs",
 "products": [
 {
 "XML_ID": "some-xml-id",
 "PHOTO": [
 "6f8b66f0-44e3-11e6-9aa8-e41f13bc5448.jpg",
 "13afe6a1-a57d-11ea-a88e-e41f13bc544a.jpg"
]
 }
]
}
```

Скрипт поддерживает отправку нескольких товаров и/или нескольких изображений, по умолчанию первое изображение и будет на сайте.

### ### Пример 1: Один товар с одним изображением

```
```json
{
  "token": "5f)m*EDs",
  "products": [
    {
      "XML_ID": "product-xml-id-1",
      "PHOTO": [
```

```

        "6f8b66f0-44e3-11e6-9aa8-e41f13bc5448.jpg"
    ]
}
]
}
...

```

Пример 2: Несколько товаров с несколькими изображением каждый

```

```json
{
 "token": "5f)m*EDs",
 "products": [
 {
 "XML_ID": "product-xml-id-1",
 "PHOTO": [
 "6f8b66f0-44e3-11e6-9aa8-e41f13bc5448.jpg",
 "13afe6a1-a57d-11ea-a88e-e41f13bc544a.jpg"
]
 },
 {
 "XML_ID": "product-xml-id-2",
 "PHOTO": [
 "6f8b66c1-44e3-11e6-9aa8-e41f13bc5448.jpg",
 "13afe69d-a57d-11ea-a88e-e41f13bc544a.jpg"
]
 }
]
}
]
}
...

```

### ### Выходные данные

Скрипт возвращает JSON-ответ с результатом выполнения:

```

```json
{
  "result": "success" | "error",
  "errmsg": ["Error message 1", "Error message 2"],
  "output": "Detailed output messages"
}
...

```