

# REST API по работе с файлами, папками и раздачей прав на них:

---

**P.S.** API используется обычное для версии **Alfresco 5.2 и выше**.

## 1. GET /api/node - Получить информацию о ноде (файл, папка)

параметры:

nodeId - Идентификатор узла. Вы также можете использовать один из этих хорошо известных псевдонимов: **-my-**, **-shared-**, **-root-**

include - Возвращает дополнительную информацию об узле. Могут быть запрошены следующие необязательные поля:

- allowableOperations
- association
- isLink
- isFavorite
- isLocked
- path
- permissions

*По умолчанию permissions*

Ответ:

```
{
  "entry": {
    "id": "string",
    "name": "string",
    "nodeType": "string",
    "isFolder": true,
    "isFile": true,
    "isLocked": true,
    "modifiedAt": "string",
    "modifiedByUser": {
      "displayName": "string",
      "id": "string"
    },
    "createdAt": "string",
    "createdByUser": {
      "displayName": "string",
      "id": "string"
    },
    "parentId": "string",
    "isLink": true,
    "isFavorite": true,
    "content": {
```

```
    "mimeType": "string",
    "mimeTypeName": "string",
    "sizeInBytes": 0,
    "encoding": "string"
  },
  "aspectNames": [
    "string"
  ],
  "properties": {},
  "allowableOperations": [
    "string"
  ],
  "path": {
    "elements": [
      {
        "id": "string",
        "name": "string",
        "nodeType": "string",
        "aspectNames": [
          "string"
        ]
      }
    ],
    "name": "string",
    "isComplete": true
  },
  "permissions": {
    "isInheritanceEnabled": true,
    "inherited": [
      {
        "authorityId": "string",
        "name": "string",
        "accessStatus": "string"
      }
    ],
    "locallySet": [
      {
        "authorityId": "string",
        "name": "string",
        "accessStatus": "string"
      }
    ],
    "settable": [
      "string"
    ]
  }
}
```

## 2. PUT /api/node - Обновить ноду (файл, папка)

Обновляет узел **nodeId**. Например, вы можете переименовать файл или папку:

```
{
  "name": "My new name"
}
```

Вы также можете установить или обновить одно, или несколько свойств:

```
{
  "properties":
  {
    "cm:title": "Folder title"
  }
}
```

**Примечание:** установка свойств типа **d: content** и **d: category** не поддерживается. Примечание: если вы хотите добавить или удалить аспекты, вы должны сначала использовать **GET /api/node**, чтобы получить полный набор **aspectNames**.

Вы можете добавить (или удалить) разрешения **localSet**, если таковые имеются, в дополнение к любым унаследованным разрешениям. Вы также можете при желании отключить (или повторно включить) унаследованные разрешения с помощью флага **isInheritanceEnabled**:

```
{
  "permissions":
  {
    "isInheritanceEnabled": false,
    "locallySet":
    [
      {"authorityId": "GROUP_special", "name": "Read",
"accessStatus": "DENIED"},
      {"authorityId": "testuser", "name": "Contributor",
"accessStatus": "ALLOWED"}
    ]
  }
}
```

Примечание. Если вы хотите добавить или удалить локально установленные разрешения, вы должны сначала использовать **GET /api/node**, чтобы получить полный набор разрешений **locallySet**.

Примечание. В настоящее время оптимистическая блокировка обновлений отсутствует, поэтому они применяются в порядке «побеждает последний».

параметры:

nodeId - Идентификатор узла.

include - Возвращает дополнительную информацию об узле. Могут быть запрошены следующие необязательные поля:

- allowableOperations
- association
- isLink
- isFavorite
- isLocked
- path
- permissions

*По умолчанию* **permissions**

Ответ:

```
{
  "name": "string",
  "nodeType": "string",
  "aspectNames": [
    "string"
  ],
  "properties": {},
  "permissions": {
    "isInheritanceEnabled": true,
    "inherited": [
      {
        "authorityId": "string",
        "name": "string",
        "accessStatus": "string"
      }
    ],
    "locallySet": [
      {
        "authorityId": "string",
        "name": "string",
        "accessStatus": "string"
      }
    ],
    "settable": [
      "string"
    ]
  }
}
```

3. **POST /api/node - Создайте узел и добавьте его в качестве основного потомка узла nodeId.**

Вы должны указать как минимум имя (**name**) и **nodeType**. Например, чтобы создать папку:

```
{
  "name": "Моя папка",
  "nodeType": "cm: folder"
}
```

Вы можете создать пустой файл следующим образом:

```
{
  "name": "Мой текстовый файл.txt",
  "nodeType": "cm: content"
}
```

Вы можете создать папку или другой узел внутри иерархии папок:

```
{
  "name": "My Special Folder",
  "nodeType": "cm: folder",
  "relativePath": "X/Y/Z"
}
```

Относительный путь (**relativePath**) определяет структуру папок для создания относительно узла **nodeId**. Папки в **relativePath**, которые не существуют, создаются до создания узла.

Вы можете установить свойства при создании нового узла:

```
{
  "name": "My Other Folder",
  "nodeType": "cm: folder",
  "properties": {
    "cm: title": "Folder title",
    "cm: description": "This is an important folder"
  }
}
```

Примечание: установка свойств типа **d: content** и **d: category** не поддерживается.

Вы также можете при желании отключить (или включить) унаследованные разрешения с помощью флага **isInheritanceEnabled**:

```
{
  "permissions": {
    "isInheritanceEnabled": false,
  }
}
```

```

    "locallySet":
      [
        {"authorityId": "GROUP_special", "name": "Read",
"accessStatus": "DENIED"},
        {"authorityId": "testuser", "name": "Contributor",
"accessStatus": "ALLOWED"}
      ]
    }
  }
}

```

Как правило, для файлов и папок основные дочерние элементы создаются в родительской папке с использованием значения по умолчанию «cm: contains» assocType. Если модель содержимого позволяет, то также можно создать первичных дочерних элементов с другим типом ассоциаций. Например:

```

{
  "name": "My Node",
  "nodeType": "my:specialNodeType",
  "association":
  {
    "assocType": "my:specialAssocType"
  }
}

```

Дополнительные ассоциации могут быть добавлены после создания узла. Вы также можете добавить ассоциации во время создания узла. Это требуется, например, если модель содержимого указывает, что узел имеет обязательные ассоциации с одним или несколькими существующими узлами. Вы можете дополнительно указать массив **secondaryChildren** для создания одной или нескольких вторичных дочерних ассоциаций, так что вновь созданный узел действует как родительский узел. Вы можете дополнительно указать массив целей для создания одной или нескольких одноранговых ассоциаций, чтобы вновь созданный узел действовал как исходный узел. Например, чтобы связать одного или нескольких вторичных потомков во время создания:

```

{
  "name": "My Folder",
  "nodeType": "cm:folder",
  "secondaryChildren":
    [ {"childId": "abcde-01234-...", "assocType": "my:specialChildAssocType"}
  ]
}

```

Например, чтобы связать одну или несколько целей во время создания:

```
{
  "name": "My Folder",
  "nodeType": "cm:folder",
  "targets":
    [ {"targetId": "abcde-01234-...", "assocType": "my:specialPeerAssocType"}
  ]
}
```

параметры:

nodeId - Идентификатор узла. Вы также можете использовать один из этих хорошо известных псевдонимов: **-my-**, **-shared-**, **-root-**

autoRename - Если **true**, то конфликт имен вызовет попытку автоматического переименования путем нахождения уникального имени с использованием целочисленного суффикса.

*По умолчанию* **false**

include - Возвращает дополнительную информацию об узле. Могут быть запрошены следующие необязательные поля:

- allowableOperations
- association
- isLink
- isFavorite
- isLocked
- path
- permissions

*По умолчанию* **path**

В теле:

```
{
  "name": "string",
  "nodeType": "string",
  "aspectNames": [
    "string"
  ],
  "properties": {},
  "permissions": {
    "isInheritanceEnabled": true,
    "inherited": [
      {
        "authorityId": "string",
        "name": "string",
```

```
        "accessStatus": "string"
      }
    ],
    "locallySet": [
      {
        "authorityId": "string",
        "name": "string",
        "accessStatus": "string"
      }
    ],
    "settable": [
      "string"
    ]
  },
  "relativePath": "string",
  "association": {
    "assocType": "string"
  },
  "secondaryChildren": [
    {
      "childId": "string",
      "assocType": "string"
    }
  ],
  "targets": [
    {
      "targetId": "string",
      "assocType": "string"
    }
  ]
}
```

Ответ:

```
{
  "entry": {
    "id": "string",
    "name": "string",
    "nodeType": "string",
    "isFolder": true,
    "isFile": true,
    "isLocked": true,
    "modifiedAt": "string",
    "modifiedByUser": {
      "displayName": "string",
      "id": "string"
    },
  },
  "createdAt": "string",
  "createdByUser": {
    "displayName": "string",
  }
}
```



```
    "id": "string"
  },
  "parentId": "string",
  "isLink": true,
  "isFavorite": true,
  "content": {
    "mimeType": "string",
    "mimeTypeName": "string",
    "sizeInBytes": 0,
    "encoding": "string"
  },
  "aspectNames": [
    "string"
  ],
  "properties": {},
  "allowableOperations": [
    "string"
  ],
  "path": {
    "elements": [
      {
        "id": "string",
        "name": "string",
        "nodeType": "string",
        "aspectNames": [
          "string"
        ]
      }
    ],
    "name": "string",
    "isComplete": true
  },
  "permissions": {
    "isInheritanceEnabled": true,
    "inherited": [
      {
        "authorityId": "string",
        "name": "string",
        "accessStatus": "string"
      }
    ],
    "locallySet": [
      {
        "authorityId": "string",
        "name": "string",
        "accessStatus": "string"
      }
    ],
    "settable": [
      "string"
    ]
  }
}
```

#### 4. DELETE /api/node - Удалить

параметры:

nodeId - Идентификатор узла.

permanent - Если **true**, то узел удаляется безвозвратно, без перемещения в корзину. Только владелец узла или администратор может удалить узел без возможности восстановления.

По умолчанию **false**

Ответ:

204

#### 5. POST /api/nodelist - Создайте список узлов, в виде "дерева" и добавьте в качестве основного потомка узла nodeId.

Вы должны указать как минимум имя (**name**) и **nodeType**. Например, чтобы создать папку:

```
{
  "name": "Моя папка",
  "nodeType": "cm: folder"
}
```

Вы можете создать пустой файл следующим образом:

```
{
  "name": "Мой текстовый файл.txt",
  "nodeType": "cm: content"
}
```

Вы можете создать папку или другой узел внутри иерархии папок:

```
{
  "name": "My Special Folder",
  "nodeType": "cm: folder",
  "relativePath": "X/Y/Z"
}
```

Относительный путь (**relativePath**) определяет структуру папок для создания относительно узла **nodeId**. Папки в **relativePath**, которые не существуют, создаются до создания узла.

Вы можете установить свойства при создании нового узла:

```
{
  "name": "My Other Folder",
  "nodeType": "cm:folder",
  "properties":
  {
    "cm:title": "Folder title",
    "cm:description": "This is an important folder"
  }
}
```

Примечание: установка свойств типа **d: content** и **d: category** не поддерживается.

Вы также можете при желании отключить (или включить) унаследованные разрешения с помощью флага `isInheritanceEnabled`:

```
{
  "permissions":
  {
    "isInheritanceEnabled": false,
    "locallySet":
    [
      {"authorityId": "GROUP_special", "name": "Read",
"accessStatus": "DENIED"},
      {"authorityId": "testuser", "name": "Contributor",
"accessStatus": "ALLOWED"}
    ]
  }
}
```

Как правило, для файлов и папок основные дочерние элементы создаются в родительской папке с использованием значения по умолчанию «cm: contains» `assocType`. Если модель содержимого позволяет, то также можно создать первичных дочерних элементов с другим типом ассоциаций. Например:

```
{
  "name": "My Node",
  "nodeType": "my:specialNodeType",
  "association":
  {
    "assocType": "my:specialAssocType"
  }
}
```

Дополнительные ассоциации могут быть добавлены после создания узла. Вы также можете добавить ассоциации во время создания узла. Это требуется, например, если модель содержимого указывает, что узел имеет обязательные ассоциации с одним или несколькими существующими узлами. Вы можете дополнительно указать массив **secondaryChildren** для создания одной или нескольких вторичных дочерних ассоциаций, так что вновь созданный узел действует как родительский узел. Вы можете дополнительно указать массив целей для создания одной или нескольких одноранговых ассоциаций, чтобы вновь созданный узел действовал как исходный узел. Например, чтобы связать одного или нескольких вторичных потомков во время создания:

```
{
  "name": "My Folder",
  "nodeType": "cm:folder",
  "secondaryChildren":
    [ {"childId": "abcde-01234-...", "assocType": "my:specialChildAssocType"} ]
}
```

Например, чтобы связать одну или несколько целей во время создания:

```
{
  "name": "My Folder",
  "nodeType": "cm:folder",
  "targets":
    [ {"targetId": "abcde-01234-...", "assocType": "my:specialPeerAssocType"} ]
}
```

параметры:

nodeId - Идентификатор узла. Вы также можете использовать один из этих хорошо известных псевдонимов: **-my-**, **-shared-**, **-root-**

autoRename - Если **true**, то конфликт имен вызовет попытку автоматического переименования путем нахождения уникального имени с использованием целочисленного суффикса.

По умолчанию **false**

include - Возвращает дополнительную информацию об узле. Могут быть запрошены следующие необязательные поля:

- allowableOperations
- association
- isLink
- isFavorite

- isLocked
- path
- permissions

*По умолчанию* **path**

В теле:

```
[
  {
    "name": "string",
    "nodeType": "string",
    "aspectNames": [
      "string"
    ],
    "properties": {},
    "permissions": {
      "isInheritanceEnabled": true,
      "inherited": [
        {
          "authorityId": "string",
          "name": "string",
          "accessStatus": "string"
        }
      ],
      "locallySet": [
        {
          "authorityId": "string",
          "name": "string",
          "accessStatus": "string"
        }
      ],
      "settable": [
        "string"
      ]
    },
    "relativePath": "string",
    "association": {
      "assocType": "string"
    },
    "secondaryChildren": [
      {
        "childId": "string",
        "assocType": "string"
      }
    ],
    "targets": [
      {
        "targetId": "string",
        "assocType": "string"
      }
    ]
  }
]
```

```
}  
]
```

Ответ:

```
{  
  "entry": {  
    "id": "string",  
    "name": "string",  
    "nodeType": "string",  
    "isFolder": true,  
    "isFile": true,  
    "isLocked": true,  
    "modifiedAt": "string",  
    "modifiedByUser": {  
      "displayName": "string",  
      "id": "string"  
    },  
    "createdAt": "string",  
    "createdByUser": {  
      "displayName": "string",  
      "id": "string"  
    },  
    "parentId": "string",  
    "isLink": true,  
    "isFavorite": true,  
    "content": {  
      "mimeType": "string",  
      "mimeTypeName": "string",  
      "sizeInBytes": 0,  
      "encoding": "string"  
    },  
    "aspectNames": [  
      "string"  
    ],  
    "properties": {},  
    "allowableOperations": [  
      "string"  
    ],  
    "path": {  
      "elements": [  
        {  
          "id": "string",  
          "name": "string",  
          "nodeType": "string",  
          "aspectNames": [  
            "string"  
          ]  
        }  
      ],  
    },  
  ],  
}
```

```

    "name": "string",
    "isComplete": true
  },
  "permissions": {
    "isInheritanceEnabled": true,
    "inherited": [
      {
        "authorityId": "string",
        "name": "string",
        "accessStatus": "string"
      }
    ],
    "locallySet": [
      {
        "authorityId": "string",
        "name": "string",
        "accessStatus": "string"
      }
    ],
    "settable": [
      "string"
    ]
  }
}

```

6. **GET /api/node/secondary-children - Получает список вторичных дочерних узлов, связанных с текущим родительским nodeId, через вторичную дочернюю ассоциацию.**

параметры:

nodeId - Идентификатор узла. Вы также можете использовать один из этих хорошо известных псевдонимов: **-my-, -shared-, -root-**

include - Возвращает дополнительную информацию об узле. Могут быть запрошены следующие необязательные поля:

- allowableOperations
- association
- isLink
- isFavorite
- isLocked
- path
- permissions

*По умолчанию path*

skipCount - Количество сущностей, которые существуют в коллекции до тех, которые включены в этот список. Если не указан, то значение по умолчанию - **0**.

maxItems - Максимальное количество элементов для возврата в список. Если не указан, то значение по умолчанию - **100**.

includeSource - Также включите источник (в дополнение к записям) с информацией о папке на **nodeId**.

*По умолчанию true*

Ответ:

```
{
  "entry": {
    "id": "string",
    "name": "string",
    "nodeType": "string",
    "isFolder": true,
    "isFile": true,
    "isLocked": true,
    "modifiedAt": "string",
    "modifiedByUser": {
      "displayName": "string",
      "id": "string"
    },
    "createdAt": "string",
    "createdByUser": {
      "displayName": "string",
      "id": "string"
    },
    "parentId": "string",
    "isLink": true,
    "isFavorite": true,
    "content": {
      "mimeType": "string",
      "mimeTypeName": "string",
      "sizeInBytes": 0,
      "encoding": "string"
    },
    "aspectNames": [
      "string"
    ],
    "properties": {},
    "allowableOperations": [
      "string"
    ],
    "path": {
      "elements": [
        {
          "id": "string",
```



```

        "name": "string",
        "nodeType": "string",
        "aspectNames": [
            "string"
        ]
    },
    ],
    "name": "string",
    "isComplete": true
},
"permissions": {
    "isInheritanceEnabled": true,
    "inherited": [
        {
            "authorityId": "string",
            "name": "string",
            "accessStatus": "string"
        }
    ],
    "locallySet": [
        {
            "authorityId": "string",
            "name": "string",
            "accessStatus": "string"
        }
    ],
    "settable": [
        "string"
    ]
}
}
}

```

7. **POST /api/node/secondary-children - Создайте вторичную дочернюю ассоциацию с заданным типом ассоциации между родительским nodeId и дочерним узлом.**

параметр:

nodeId - Идентификатор узла.

В теле:

```

{
    "childId": "string",
    "assocType": "string"
}

```

Ответ:

```
{
  "entry": {
    "id": "string",
    "name": "string",
    "nodeType": "string",
    "isFolder": true,
    "isFile": true,
    "isLocked": true,
    "modifiedAt": "string",
    "modifiedByUser": {
      "displayName": "string",
      "id": "string"
    },
    "createdAt": "string",
    "createdByUser": {
      "displayName": "string",
      "id": "string"
    },
    "parentId": "string",
    "isLink": true,
    "isFavorite": true,
    "content": {
      "mimeType": "string",
      "mimeTypeName": "string",
      "sizeInBytes": 0,
      "encoding": "string"
    },
    "aspectNames": [
      "string"
    ],
    "properties": {},
    "allowableOperations": [
      "string"
    ],
    "path": {
      "elements": [
        {
          "id": "string",
          "name": "string",
          "nodeType": "string",
          "aspectNames": [
            "string"
          ]
        }
      ],
      "name": "string",
      "isComplete": true
    },
    "permissions": {
      "isInheritanceEnabled": true,
      "inherited": [
        {
          "authorityId": "string",
```

```

        "name": "string",
        "accessStatus": "string"
      }
    ],
    "locallySet": [
      {
        "authorityId": "string",
        "name": "string",
        "accessStatus": "string"
      }
    ],
    "settable": [
      "string"
    ]
  }
}

```

**8. POST /api/node/secondary-children/list - Создайте более одной вторичной дочерней ассоциации с заданным типом ассоциации между родительским nodeId и дочерним узлом.**

Примечание. Вы можете создать более одной вторичной дочерней ассоциации, указав список ассоциаций в теле JSON следующим образом:

```

[
  {
    "childId": "string",
    "assocType": "string"
  },
  {
    "childId": "string",
    "assocType": "string"
  }
]

```

Если вы укажете список в качестве ввода, то в теле ответа будет возвращен список с разбивкой на страницы, а не запись. Например:

```

{
  "list": {
    "pagination": {
      "count": 2,
      "hasMoreItems": false,
      "totalItems": 2,
      "skipCount": 0,
      "maxItems": 100
    },
    "entries": [
      {

```

```
    "entry": {  
    }  
  },  
  {  
    "entry": {  
    }  
  }  
]  
}  
}
```

параметр:

nodeId - Идентификатор узла.

В теле:

```
[  
  {  
    "childId": "string",  
    "assocType": "string"  
  }  
]
```

Ответ:

```
{  
  "entry": {  
    "id": "string",  
    "name": "string",  
    "nodeType": "string",  
    "isFolder": true,  
    "isFile": true,  
    "isLocked": true,  
    "modifiedAt": "string",  
    "modifiedByUser": {  
      "displayName": "string",  
      "id": "string"  
    },  
    "createdAt": "string",  
    "createdByUser": {  
      "displayName": "string",  
      "id": "string"  
    },  
    "parentId": "string",  
    "isLink": true,  
    "isFavorite": true,  
  },  
}
```

```
"content": {
  "mimeType": "string",
  "mimeTypeName": "string",
  "sizeInBytes": 0,
  "encoding": "string"
},
"aspectNames": [
  "string"
],
"properties": {},
"allowableOperations": [
  "string"
],
"path": {
  "elements": [
    {
      "id": "string",
      "name": "string",
      "nodeType": "string",
      "aspectNames": [
        "string"
      ]
    }
  ],
  "name": "string",
  "isComplete": true
},
"permissions": {
  "isInheritanceEnabled": true,
  "inherited": [
    {
      "authorityId": "string",
      "name": "string",
      "accessStatus": "string"
    }
  ],
  "locallySet": [
    {
      "authorityId": "string",
      "name": "string",
      "accessStatus": "string"
    }
  ],
  "settable": [
    "string"
  ]
}
}
```

## 9. DELETE /api/node/secondary-children - Удалить второстепенный дочерний или дочерние ноды

Удалите вторичные дочерние ассоциации между родительским `nodeId` и дочерними узлами для данного типа ассоциации. Если тип ассоциации не указан, то все вторичные дочерние ассоциации любого типа в направлении от родительского к вторичному дочернему элементу будут удалены. У ребенка по-прежнему будет основной родитель, и он может быть связан как дополнительный ребенок с другими дополнительными родителями.

параметры:

`nodeId` - Идентификатор родительского узла.

`childId` - Идентификатор дочернего узла.

Ответ:

204

#### 10. [POST /api/node/move](#) - Переместите узел.

Переместите узел **`nodeId`** в узел родительской папки **`targetParentId`**. **`**targetParentId`** указывается в теле запроса. Перемещенный узел сохраняет свое имя, если вы не укажете новое имя в теле запроса. Если исходный **`nodeId`** - это папка, то ее дочерние элементы также перемещаются. Этот ход фактически изменит основного родителя.

параметры:

`nodeId` - Идентификатор узла.

`include` - Возвращает дополнительную информацию об узле. Могут быть запрошены следующие необязательные поля:

- `allowableOperations`
- `association`
- `isLink`
- `isFavorite`
- `isLocked`
- `path`
- `permissions`

*По умолчанию **`path`***

В теле:

```
{
  "targetParentId": "string",
```

```
"name": "string"
}
```

Ответ:

```
{
  "entry": {
    "id": "string",
    "name": "string",
    "nodeType": "string",
    "isFolder": true,
    "isFile": true,
    "isLocked": true,
    "modifiedAt": "string",
    "modifiedByUser": {
      "displayName": "string",
      "id": "string"
    },
    "createdAt": "string",
    "createdByUser": {
      "displayName": "string",
      "id": "string"
    },
    "parentId": "string",
    "isLink": true,
    "isFavorite": true,
    "content": {
      "mimeType": "string",
      "mimeTypeName": "string",
      "sizeInBytes": 0,
      "encoding": "string"
    },
    "aspectNames": [
      "string"
    ],
    "properties": {},
    "allowableOperations": [
      "string"
    ],
    "path": {
      "elements": [
        {
          "id": "string",
          "name": "string",
          "nodeType": "string",
          "aspectNames": [
            "string"
          ]
        }
      ]
    }
  },
}
```

```

        "name": "string",
        "isComplete": true
    },
    "permissions": {
        "isInheritanceEnabled": true,
        "inherited": [
            {
                "authorityId": "string",
                "name": "string",
                "accessStatus": "string"
            }
        ],
        "locallySet": [
            {
                "authorityId": "string",
                "name": "string",
                "accessStatus": "string"
            }
        ],
        "settable": [
            "string"
        ]
    }
}

```

#### 11. [POST /api/node/copy](#) - Скопировать узел.

Копирует узел **nodeId** в узел **targetParentId** родительской папки. Вы указываете **targetParentId** в теле запроса. Новый узел имеет то же имя, что и исходный узел, если вы не укажете новое имя в теле запроса. Если исходный **nodeId** - это папка, то все ее дочерние элементы также копируются. Если исходный **nodeId** является файлом, его свойства, аспекты и теги копируются, а рейтинги, комментарии и блокировки - нет.

параметры:

**nodeId** - Идентификатор узла.

**include** - Возвращает дополнительную информацию об узле. Могут быть запрошены следующие необязательные поля:

- allowableOperations
- association
- isLink
- isFavorite
- isLocked
- path
- permissions

*По умолчанию path*



В теле:

```
{
  "targetParentId": "string",
  "name": "string"
}
```

Ответ:

```
{
  "entry": {
    "id": "string",
    "name": "string",
    "nodeType": "string",
    "isFolder": true,
    "isFile": true,
    "isLocked": true,
    "modifiedAt": "string",
    "modifiedByUser": {
      "displayName": "string",
      "id": "string"
    },
    "createdAt": "string",
    "createdByUser": {
      "displayName": "string",
      "id": "string"
    },
    "parentId": "string",
    "isLink": true,
    "isFavorite": true,
    "content": {
      "mimeType": "string",
      "mimeTypeName": "string",
      "sizeInBytes": 0,
      "encoding": "string"
    },
    "aspectNames": [
      "string"
    ],
    "properties": {},
    "allowableOperations": [
      "string"
    ],
    "path": {
      "elements": [
        {
          "id": "string",
          "name": "string",

```

```

        "nodeType": "string",
        "aspectNames": [
            "string"
        ]
    },
    ],
    "name": "string",
    "isComplete": true
},
"permissions": {
    "isInheritanceEnabled": true,
    "inherited": [
        {
            "authorityId": "string",
            "name": "string",
            "accessStatus": "string"
        }
    ],
    "locallySet": [
        {
            "authorityId": "string",
            "name": "string",
            "accessStatus": "string"
        }
    ],
    "settable": [
        "string"
    ]
}
}
}
}

```

## 12. GET /api/node/parents - Получить список родителей

Получает список родительских узлов, связанных с текущим дочерним **nodeId**. В список входят как основной родитель, так и все второстепенные родители.

параметры:

**nodeId** - Идентификатор узла. Вы также можете использовать один из этих хорошо известных псевдонимов: **-my-**, **-shared-**, **-root-**

**where** - При желании отфильтруйте список по **assocType** и/или **isPrimary**. Вот несколько примеров фильтров:

- where=(assocType='my:specialAssocType')
- where=(isPrimary=true)
- where=(isPrimary=false and assocType='my:specialAssocType')

*По умолчанию isPrimary=true*

include - Возвращает дополнительную информацию об узле. Могут быть запрошены следующие необязательные поля:

- allowableOperations
- association
- isLink
- isFavorite
- isLocked
- path
- permissions

*По умолчанию path*

skipCount - Количество сущностей, которые существуют в коллекции до тех, которые включены в этот список. Если не указан, то значение по умолчанию - **0**.

maxItems - Максимальное количество элементов для возврата в список. Если не указан, то значение по умолчанию - **100**.

includeSource - Также включите источник (в дополнение к записям) с информацией о папке на **nodeId**.

*По умолчанию true*

Ответ:

```
{
  "entry": {
    "id": "string",
    "name": "string",
    "nodeType": "string",
    "isFolder": true,
    "isFile": true,
    "isLocked": true,
    "modifiedAt": "string",
    "modifiedByUser": {
      "displayName": "string",
      "id": "string"
    },
    "createdAt": "string",
    "createdByUser": {
      "displayName": "string",
      "id": "string"
    },
    "parentId": "string",
    "isLink": true,
```

```
    "isFavorite": true,
    "content": {
      "mimeType": "string",
      "mimeTypeName": "string",
      "sizeInBytes": 0,
      "encoding": "string"
    },
    "aspectNames": [
      "string"
    ],
    "properties": {},
    "allowableOperations": [
      "string"
    ],
    "path": {
      "elements": [
        {
          "id": "string",
          "name": "string",
          "nodeType": "string",
          "aspectNames": [
            "string"
          ]
        }
      ],
      "name": "string",
      "isComplete": true
    },
    "permissions": {
      "isInheritanceEnabled": true,
      "inherited": [
        {
          "authorityId": "string",
          "name": "string",
          "accessStatus": "string"
        }
      ],
      "locallySet": [
        {
          "authorityId": "string",
          "name": "string",
          "accessStatus": "string"
        }
      ],
      "settable": [
        "string"
      ]
    }
  }
}
```

### 13. GET /api/node/children - Получить список дочерних узлов

Получаем список дочерних узлов родительского узла **nodeId**. По умолчанию возвращается минимальная информация для каждого ребенка.

Вы можете использовать параметр **include** для возврата дополнительной информации.

Список дочерних узлов включает первичные и вторичные дочерние узлы, если таковые имеются.

Вы можете использовать параметр **include (include = association)**, чтобы вернуть детали дочерней ассоциации для каждого дочернего элемента, включая **assocType** и флаг **isPrimary**.

Порядок сортировки по умолчанию для возвращенного списка - это сортировка папок перед файлами и по возрастанию имени.

Вы можете переопределить значение по умолчанию, используя **orderBy**, чтобы указать одно или несколько полей для сортировки. По умолчанию порядок сортировки всегда возрастающий, но вы можете использовать дополнительный модификатор **ASC** или **DESC**, чтобы указать порядок сортировки по возрастанию или убыванию. Например, указание **orderBy = name DESC** возвращает смешанный список папок/файлов в порядке убывания имен.

Вы можете использовать любое из следующих полей, чтобы упорядочить результаты:

- isFolder
- name
- mimeType
- nodeType
- sizeInBytes
- modifiedAt
- createdAt
- modifiedByUser
- createdByUser

параметры:

**nodeId** - Идентификатор узла. Вы также можете использовать один из этих хорошо известных псевдонимов: **-my-**, **-shared-**, **-root-**

**skipCount** - Количество сущностей, которые существуют в коллекции до тех, которые включены в этот список. Если не указан, то значение по умолчанию - **0**.

**maxItems** - Максимальное количество элементов для возврата в список. Если не указан, то значение по умолчанию - **100**.

**orderBy** - Строка для управления порядком объектов, возвращаемых в списке. Вы можете использовать параметр **orderBy** для сортировки списка по одному или нескольким полям. Каждое поле имеет порядок сортировки по умолчанию, который обычно является возрастающим. Чтобы отсортировать объекты в определенном порядке, вы можете использовать ключевые слова **ASC** и **DESC** для любого поля.

### *По умолчанию* **ASC**

where - При желании отфильтруйте список. Вот некоторые примеры:

- where=(isFolder=true)
- where=(isFile=true)
- where=(nodeType='my:specialNodeType')
- where=(nodeType='my:specialNodeType INCLUDESUBTYPES')
- where=(isPrimary=true)
- where=(assocType='my:specialAssocType')
- where=(isPrimary=false and assocType='my:specialAssocType')

### *По умолчанию* **isFolder=true**

include - Возвращает дополнительную информацию об узле. Могут быть запрошены следующие необязательные поля:

- allowableOperations
- association
- isLink
- isFavorite
- isLocked
- path
- permissions

### *По умолчанию* **path,permissions**

includeSource - Также включите источник в дополнение к записям с информацией о папке на родительском узле - либо с указанным родительским **nodeId**, либо с разрешенным с помощью **relativePath**.

### *По умолчанию* **true**

Ответ:

```
{
  "entry": {
    "id": "string",
    "name": "string",
    "nodeType": "string",
    "isFolder": true,
    "isFile": true,
    "isLocked": true,
    "modifiedAt": "string",
```

```
"modifiedByUser": {
  "displayName": "string",
  "id": "string"
},
"createdAt": "string",
"createdByUser": {
  "displayName": "string",
  "id": "string"
},
"parentId": "string",
"isLink": true,
"isFavorite": true,
"content": {
  "mimeType": "string",
  "mimeTypeName": "string",
  "sizeInBytes": 0,
  "encoding": "string"
},
"aspectNames": [
  "string"
],
"properties": {},
"allowableOperations": [
  "string"
],
"path": {
  "elements": [
    {
      "id": "string",
      "name": "string",
      "nodeType": "string",
      "aspectNames": [
        "string"
      ]
    }
  ],
  "name": "string",
  "isComplete": true
},
"permissions": {
  "isInheritanceEnabled": true,
  "inherited": [
    {
      "authorityId": "string",
      "name": "string",
      "accessStatus": "string"
    }
  ],
  "locallySet": [
    {
      "authorityId": "string",
      "name": "string",
      "accessStatus": "string"
    }
  ]
}
```

```
    ],  
    "settable": [  
      "string"  
    ]  
  }  
}
```