**`InventoryManagement API V1 documentation**

InventoryManagement API describes the work with API in InventoryManagement.

**Request and response formats**

In general, the InventoryManagement API uses HTTP POST requests with JSON arguments and JSON responses.

**Browser-based JavaScript, CSRF**

InventoryManagement don`t allow using CSRF – anti-CSRF is part of of InventoryManagement API engine.

API engine uses URL parameters.

The Content-Type is "application/json; charset=utf-8". Except “/Token”.

**Date format**

All dates in the API are strings in the [ISO 8601 "combined date and time representation" format](https://en.wikipedia.org/wiki/ISO_8601#Combined_date_and_time_representations).

**Path formats**

Paths are relative to an application's identities (api/Goods, api/Transactions, api/TransactionTypes, …).

**Default functions**

Every entity has default set of operations realized by CRUD set.

**Routing tables**

Here are some possible HTTP requests, along with the action that gets invoked for each:

|  |  |  |
| --- | --- | --- |
| HTTP Method | URI Path | Action |
| GET | /api/{entityController}(/id) | Get all entities in list (Read details of the exact entity) |
| POST | /api/{entityController} | Create new entity |
| DELETE | /api/{entityController}/id | Delete the entity with set id |
| PUT | /api/{entityController} | Update the entity with new variation of entity |

**Error handling**

Errors are returned using standard HTTP error code syntax. Any additional info is included as JSON in the body of the response.

**Authorization**

InventoryManagement supports OAuth 2.0 for authorizing API requests. Authorized requests to the API should use an Authorization header with the value Bearer <TOKEN>, where <TOKEN> is an access token obtained through the OAuth flow.

|  |  |
| --- | --- |
| **/api/Account/Register** | |
| DESCRIPTION | API for registering new user in system. Uses standard Identity code. |
| URL STRUCTURE | /api/Account/Register |
| METHOD | POST |
| PARAMETERS | [FromURI]  -  [FromBody]  name: Email, type: string. Login name for user.  name: Password, type: string. Password for user.  name: ConfirmPassword, type: string. Password doubling.  Sample: {"Email": "3@2.com", "Password": "oStwNfeo8T ", "ConfirmPassword": "oStwNfeo8T "} |
| RETURNS | HTTP code 200 - at success,  HTTP code 404 - at fail. |
| ERRORS | Returns standard Identity errors. |

|  |  |
| --- | --- |
| **/token** | |
| DESCRIPTION | API for authorizing user in system. Uses standard Identity code. |
| URL STRUCTURE | /token |
| METHOD | POST |
| Content-Type | application/x-www-form-urlencoded |
| PARAMETERS | [FromURI]  -  [FromBody]  name: grant\_type, type: string. Only value is “password”.  name: [username](mailto:username=3@2.com), type: string. Users name.  name: password, type: string. Users password.  Sample: [grant\_type=password&username=3@2.com&password=oStwNfeo8T](mailto:grant_type=password&username=3@2.com&password=oStwNfeo8T) |
| RETURNS | Token, type: Bearer. |
| ERRORS | Returns standard Identity errors. |

**Goods**

Goods entity is one of the main entities in application – its API controller represents CRUD functionality for work with each entity of goods and getting current quantity of each good.

|  |  |
| --- | --- |
| **/api/Goods** | |
| DESCRIPTION | Gets the list of current Users Goods, according availability for logged user role |
| URL STRUCTURE | api/Goods |
| METHOD | GET |
| PARAMETERS | API supports OData client-side paging. First of all, works orderby & direction. Skip&top – at the end.  [FromURI]  name: orderby, type: String?. Sets the column sort order for query.  name: direction, type: String?. Sets the direction sort for query. Default: asc. Available: desc.  name: skip, type: int?. Comes in pair with top. Sets number of skipped elements in query for return.  name: top, type: int?. Sets number of displayed elements in query for return.  [FromBody]  - |
| RETURNS | type: JSON-encoded List<Goods>. Returns list of all requested entities, available to this User. Sample response:  [{"Sizes":{"Id":1,"Size":"40\*60\*20"},"TypeOfStorage":{"Id":1,"Type":"Palette"},"Id":1,"Name":"Salt","Quantity":10,"TypeOfStorageId":1,"SizesId":1, "Info":"1 good info"}, {"Sizes":{"Id":1, "Size":"40\*60\*20"}, "TypeOfStorage":{"Id":1, "Type":"Palette"}, "Id":2, "Name":"Pepper", "Quantity":5, "TypeOfStorageId":1, "SizesId":1, "Info":"2 info"}] |
| ERRORS | 400 – Not valid parameters (OData, JSON)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/Goods/id** | |
| DESCRIPTION | Gets one exact entity of Transaction of logged User, according availability for logged user role. |
| URL STRUCTURE | /api/Goods/id |
| METHOD | GET |
| PARAMETERS | [FromURI]  name: Id, type: int. Represents Id of targeted entity  [FromBody]  - |
| RETURNS | type: JSON. One entity of targeted entity if it`s available to this User. Sample response: [{"Sizes":{"Id":1,"Size":"40\*60\*20"},"TypeOfStorage":{"Id":1,"Type":"Palette"}, "Id":1,"Name":"Salt","Quantity":10,"TypeOfStorageId":1,"SizesId":1, "Info":"1 good info"}] |
| ERRORS | 400 – Not valid parameters (Id, OData)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/Goods/All** | |
| DESCRIPTION | This action is available only for administrators. Gets full list of Goods. |
| URL STRUCTURE | /api/Goods/All |
| METHOD | GET` |
| PARAMETERS | [FromURI]  name: orderby, type: String?. Sets the column sort order for query.  name: direction, type: String?. Sets the direction sort for query. Default: asc. Available: desc.  name: skip, type: int?. Comes in pair with top. Sets number of skipped elements in query for return.  name: top, type: int?. Sets number of displayed elements in query for return.  [FromBody]  - |
| RETURNS | type: JSON. List of Goods. Sample of response – same as : /api/Goods (GET). |
| ERRORS | 400 – Not valid parameters (OData,)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/Goods** | |
| DESCRIPTION | Sends new entity of Goods to DB, according availability for logged user role |
| URL STRUCTURE | /api/Goods |
| METHOD | POST |
| PARAMETERS | [FromURI]  -  [FromBody]  name: good, type: JSON-encoded <GoodDTO>. One requested entity. Sample: [{"Name":"Salt", " TypeOfStorageId":1, " SizesId":1, " Info":"Some optional info", "Id":1}] |
| RETURNS | HTTP code 200 - at successful insert in DB,  HTTP code 500 - at fail. |
| ERRORS | 400 – Not valid parameters (JSON)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/Goods/id** | |
| DESCRIPTION | Deletes one exact entity of Goods in DB, according availability for logged user role |
| URL STRUCTURE | [/api/Goods/id](http://localhost/EFlocal/api/Goods/id) |
| METHOD | DELETE |
| PARAMETERS | [FromURI]  name: Id, type: string. Represents Id of targeted entity  [FromBody]  - |
| RETURNS | HTTP code 200 - at successful insert in DB,  HTTP code 500 - at fail. |
| ERRORS | 400 – Not valid parameters (Id, JSON)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/Goods** | |
| DESCRIPTION | Updates one exact entity of Goods in DB, according availability for logged user role |
| URL STRUCTURE | /api/Goods |
| METHOD | PUT |
| PARAMETERS | [FromURI]  -  [FromBody]  name: good, type: JSON-encoded GoodDTO. One entity of updated entity. Sample: same as /api/Goods (POST). |
| RETURNS | HTTP code 200 - at successful insert in DB,  HTTP code 500 - at fail. |
| ERRORS | 400 – Not valid parameters (JSON)  401 – User is not authorized  500 – Error at work with DB |

**GoodsInTransaction**

GoodsInTransaction API controller represents CRUD functionality for work with details for Transactions – which the Goods are in exact Transaction and the quantity of ones.

|  |  |
| --- | --- |
| **/api/GoodsInTransaction** | |
| DESCRIPTION | Gets the list of all GoodsInTransaction, according availability for logged user role |
| URL STRUCTURE | /api/GoodsInTransaction |
| METHOD | GET |
| PARAMETERS | API supports OData client-side paging. First of all, works orderby & direction. Skip&top – at the end.  [FromURI]  name: orderby, type: String?. Sets the column sort order for query.  name: direction, type: String?. Sets the direction sort for query. Default: asc. Available: desc.  name: skip, type: int?. Comes in pair with top. Sets number of skipped elements in query for return.  name: top, type: int?. Sets number of displayed elements in query for return.  [FromBody]  - |
| RETURNS | type: JSON-encoded List<GoodsInTransaction>. Returns list of all requested entities, available to this User. Sample response: [{"Goods":{"Sizes":{"Id":1,"Size":"40\*60\*20"},"TypeOfStorage":{"Id":1,"Type":"Palette"},"Id":1,"Name":"Salt","Quantity":10,"TypeOfStorageId":1,"SizesId":1,"Info":"1 good info"}, "Transactions": {"TranactionType": {"Id":2, "Name":"Order"}, "Users":{"Email":"1@2.com", "UserName":"1@2.com", "Id":1}, "Description":"1st transaction", "TranactionTypeId":2, "UserId":1,"Date":"2016-05-30T00:00:00", "Id":1},"Id":1, "Quantity":2, "TransactionsId":1, "GoodsId":1}, {"Goods":{"Sizes":{"Id":1, "Size":"40\*60\*20"}, "TypeOfStorage":{"Id":1, "Type":"Palette"}, "Id":2, "Name":"Pepper", "Quantity":5, "TypeOfStorageId":1, "SizesId":1, "Info":"2 info"}, "Transactions":{"TranactionType":{"Id":2, "Name":"Order"}, "Users":{"Email":"1@2.com", "UserName":"1@2.com", "Id":1}, "Description":"1st transaction", "TranactionTypeId":2, "UserId":1, "Date":"2016-05-30T00:00:00", "Id":1}, "Id":2, "Quantity":3, "TransactionsId":1, "GoodsId":2}] |
| ERRORS | 400 – Not valid parameters (OData, JSON)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/GoodsInTransaction/id** | |
| DESCRIPTION | Gets one exact entity of GoodsInTransaction entity, according availability for logged user role. |
| URL STRUCTURE | /api/GoodsInTransaction/id |
| METHOD | GET |
| PARAMETERS | [FromURI]  name: Id, type: int. Represents Id of targeted entity  [FromBody]  - |
| RETURNS | type: JSON. One entity of targeted entity if it`s available to this User. Sample response: [{"Goods":{"Sizes":{"Id":1,"Size":"40\*60\*20"},"TypeOfStorage":{"Id":1,"Type":"Palette"},"Id":1,"Name":"Salt","Quantity":10,"TypeOfStorageId":1,"SizesId":1,"Info":"1 good info"}, "Transactions": {"TranactionType": {"Id":2, "Name":"Order"}, "Users":{"Email":"1@2.com", "UserName":"1@2.com", "Id":1}, "Description":"1st transaction", "TranactionTypeId":2, "UserId":1,"Date":"2016-05-30T00:00:00", "Id":1},"Id":1, "Quantity":2, "TransactionsId":1, "GoodsId":1}] |
| ERRORS | 400 – Not valid parameters (Id, OData)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/GoodsInTransaction/All** | |
| DESCRIPTION | This action is available only for administrators. Gets full list of GoodsInTransaction. |
| URL STRUCTURE | /api/GoodsInTransaction/All |
| METHOD | GET` |
| PARAMETERS | [FromURI]  name: orderby, type: String?. Sets the column sort order for query.  name: direction, type: String?. Sets the direction sort for query. Default: asc. Available: desc.  name: skip, type: int?. Comes in pair with top. Sets number of skipped elements in query for return.  name: top, type: int?. Sets number of displayed elements in query for return.  [FromBody]  - |
| RETURNS | type: JSON. List of Transactions. Sample of response – same as : /api/GoodsInTransaction (GET). |
| ERRORS | 400 – Not valid parameters (OData,)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/GoodsInTransaction** | |
| DESCRIPTION | Sends new entity of GoodsInTransaction to DB, according availability for logged user role |
| URL STRUCTURE | /api/GoodsInTransaction |
| METHOD | POST |
| PARAMETERS | name: goodsInTransaction, type: JSON-encoded < GoodsInTransactionDTO>. One requested entity. Sample: [{"Id":1, "Quantity":2, "TransactionsId":1, "GoodsId":1}] |
| RETURNS | At success – HTTP code 200  At fail – HTTP code 400 |
| ERRORS | 400 – Failed to insert in DB  401 – User is not authorized  403 – CSRF check is not passed  415 – JSON request is not valid  503 – DB is unreachable |

|  |  |
| --- | --- |
| **/api/GoodsInTransaction/id** | |
| DESCRIPTION | Deletes one exact entity of GoodsInTransaction in DB, according availability for logged user role |
| URL STRUCTURE | [/api/GoodsInTransaction/id](http://localhost/EFlocal/api/GoodsInTransaction/id) |
| METHOD | DELETE |
| PARAMETERS | [FromURI]  name: Id, type: int. Represents Id of targeted entity  [FromBody]  - |
| RETURNS | HTTP code 200 - at successful insert in DB,  HTTP code 500 - at fail. |
| ERRORS | 400 – Not valid parameters (Id, JSON)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/GoodsInTransaction** | |
| DESCRIPTION | Updates one exact entity of GoodsInTransaction in DB, according availability for logged user role |
| URL STRUCTURE | /api/GoodsInTransaction |
| METHOD | PUT |
| PARAMETERS | [FromURI]  -  [FromBody]  name: goodsInTransaction, type: JSON-encoded GoodsInTransactionDTO. One entity of updated entity. Sample: same as /api/GoodsInTransaction (POST). |
| RETURNS | HTTP code 200 - at successful insert in DB,  HTTP code 500 - at fail. |
| ERRORS | 400 – Not valid parameters (JSON)  401 – User is not authorized  500 – Error at work with DB |

**GoodsInWarehauses**

GoodsInWarehauses API controller represents CRUD functionality for work with location of Goods – where the exact Goods are located.

|  |  |
| --- | --- |
| **/api/GoodsInWarehauses** | |
| DESCRIPTION | Gets the list of all GoodsInWarehauses, according availability for logged user role |
| URL STRUCTURE | /api/GoodsInWarehauses |
| METHOD | GET |
| PARAMETERS | API supports OData client-side paging. First of all, works orderby & direction. Skip&top – at the end.  [FromURI]  name: orderby, type: String?. Sets the column sort order for query.  name: direction, type: String?. Sets the direction sort for query. Default: asc. Available: desc.  name: skip, type: int?. Comes in pair with top. Sets number of skipped elements in query for return.  name: top, type: int?. Sets number of displayed elements in query for return.  [FromBody]  - |
| RETURNS | type: JSON-encoded List<GoodsInWarehauses>. Returns list of all requested entities, available to this User. Sample response: [{"Goods":{"Sizes":{"Id":1,"Size":"40\*60\*20"},"TypeOfStorage":{"Id":1,"Type":"Palette"},"Id":1,"Name":"Salt","Quantity":10,"TypeOfStorageId":1,"SizesId":1,"Info":"1 good info"}, "WarehousesPlaces":{"Id":1, "Adress":"Kharkiv", "Place":"1"}, "Id":2,"GoodsId":1,"WarehousesPlacesId":1}, {"Goods":{"Sizes":{"Id":1, "Size":"40\*60\*20"}, "TypeOfStorage":{"Id":1, "Type":"Palette"}, "Id":2,"Name":"Pepper", "Quantity":5, "TypeOfStorageId":1, "SizesId":1,"Info":"2 info"}, "WarehousesPlaces":{"Id":1, "Adress":"Kharkiv", "Place":"1"},"Id":3,"GoodsId":2,"WarehousesPlacesId":1}] |
| ERRORS | 400 – Not valid parameters (OData, JSON)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/GoodsInWarehauses/id** | |
| DESCRIPTION | Gets one exact entity of GoodsInWarehauses, according availability for logged user role |
| URL STRUCTURE | /api/GoodsInWarehauses/id |
| METHOD | GET |
| PARAMETERS | [FromURI]  name: Id, type: int. Represents Id of targeted entity  [FromBody]  - |
| RETURNS | type: JSON. One entity if it`s available to this User. Sample response: [{"Goods":{"Sizes":{"Id":1,"Size":"40\*60\*20"},"TypeOfStorage":{"Id":1,"Type":"Palette"},"Id":1,"Name":"Salt","Quantity":10,"TypeOfStorageId":1,"SizesId":1,"Info":"1 good info"}, "WarehousesPlaces":{"Id":1, "Adress":"Kharkiv", "Place":"1"}, "Id":2,"GoodsId":1,"WarehousesPlacesId":1}] |
| ERRORS | 400 – Not valid parameters (Id, OData)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/GoodsInWarehauses/All** | |
| DESCRIPTION | This action is available only for administrators. Gets full list of GoodsInWarehauses. |
| URL STRUCTURE | /api/GoodsInWarehauses/All |
| METHOD | GET` |
| PARAMETERS | [FromURI]  name: orderby, type: String?. Sets the column sort order for query.  name: direction, type: String?. Sets the direction sort for query. Default: asc. Available: desc.  name: skip, type: int?. Comes in pair with top. Sets number of skipped elements in query for return.  name: top, type: int?. Sets number of displayed elements in query for return.  [FromBody]  - |
| RETURNS | type: JSON. List of GoodsInWarehauses. Sample of response – same as : /api/GoodsInWarehauses (GET). |
| ERRORS | 400 – Not valid parameters (OData,)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/GoodsInWarehauses** | |
| DESCRIPTION | Sends new entity of GoodsInWarehauses to DB, according availability for logged user role |
| URL STRUCTURE | /api/GoodsInWarehauses |
| METHOD | POST |
| PARAMETERS | [FromURI]  -  [FromBody]  name: goodsInWarehauses, type: JSON-encoded < GoodsInWarehausesDTO>. One requested entity. Sample: [{"Id":2,"GoodsId":1,"WarehousesPlacesId":1}] |
| RETURNS | HTTP code 200 - at successful insert in DB,  HTTP code 500 - at fail. |
| ERRORS | 400 – Not valid parameters (JSON)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/GoodsInWarehauses/id** | |
| DESCRIPTION | Deletes one exact entity of GoodsInWarehauses in DB, according availability for logged user role |
| URL STRUCTURE | [/api/GoodsInWarehauses/id](http://localhost/EFlocal/api/GoodsInWarehauses/id) |
| METHOD | DELETE |
| PARAMETERS | [FromURI]  name: Id, type: int. Represents Id of targeted entity  [FromBody]  - |
| RETURNS | HTTP code 200 - at successful insert in DB,  HTTP code 500 - at fail. |
| ERRORS | 400 – Not valid parameters (Id, JSON)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/GoodsInWarehauses** | |
| DESCRIPTION | Updates one exact entity of GoodsInWarehauses in DB, according availability for logged user role |
| URL STRUCTURE | /api/GoodsInWarehauses |
| METHOD | PUT |
| PARAMETERS | [FromURI]  -  [FromBody]  name: goodsInWarehauses, type: JSON-encoded TransactionDTO. One entity of updated entity. Sample: same as /api/GoodsInWarehauses (POST). |
| RETURNS | HTTP code 200 - at successful insert in DB,  HTTP code 500 - at fail. |
| ERRORS | 400 – Not valid parameters (JSON)  401 – User is not authorized  500 – Error at work with DB |

**Restrictions**

Restrictions is one of the supporting entities in application – its API controller represents CRUD functionality for work with dictionary of goods restrictions.

|  |  |
| --- | --- |
| **/api/Restrictions** | |
| DESCRIPTION | Gets the list of all Restrictions, according availability for logged user role |
| URL STRUCTURE | /api/Restrictions |
| METHOD | GET |
| PARAMETERS | API supports OData client-side paging. First of all, works orderby & direction. Skip&top – at the end.  [FromURI]  name: orderby, type: String?. Sets the column sort order for query.  name: direction, type: String?. Sets the direction sort for query. Default: asc. Available: desc.  name: skip, type: int?. Comes in pair with top. Sets number of skipped elements in query for return.  name: top, type: int?. Sets number of displayed elements in query for return.  [FromBody]  - |
| RETURNS | type: JSON-encoded List<Restrictions>. Returns list of all requested entities, available to this User. Sample response: [{"Id":1,"RestrictionName":"Fragile"},{"Id":2,"RestrictionName":"Flameable"}] |
| ERRORS | 400 – Not valid parameters (OData, JSON)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/Restrictions/id** | |
| DESCRIPTION | Gets one exact entity of Restriction, according availability for logged user role |
| URL STRUCTURE | /api/Restrictions/id |
| METHOD | GET |
| PARAMETERS | [FromURI]  name: Id, type: int. Represents Id of targeted entity  [FromBody]  - |
| RETURNS | type: JSON. One entity if it`s available to this User. Sample response: [{"Id":1,"RestrictionName":"Fragile"}] |
| ERRORS | 400 – Not valid parameters (Id, OData)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/Restrictions/All** | |
| DESCRIPTION | This action is available only for administrators. Gets full list of Restrictions. |
| URL STRUCTURE | /api/Restrictions/All |
| METHOD | GET` |
| PARAMETERS | [FromURI]  name: orderby, type: String?. Sets the column sort order for query.  name: direction, type: String?. Sets the direction sort for query. Default: asc. Available: desc.  name: skip, type: int?. Comes in pair with top. Sets number of skipped elements in query for return.  name: top, type: int?. Sets number of displayed elements in query for return.  [FromBody]  - |
| RETURNS | type: JSON. List of Restrictions. Sample of response – same as : /api/Restrictions (GET). |
| ERRORS | 400 – Not valid parameters (OData,)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/Restrictions** | |
| DESCRIPTION | Sends new entity of Restrictions to DB, according availability for logged user role |
| URL STRUCTURE | /api/Restrictions |
| METHOD | POST |
| PARAMETERS | [FromURI]  -  [FromBody]  name: restrictions, type: JSON-encoded < RestrictionsDTO>. One requested entity. Sample: [{"Id":1,"RestrictionName":"Fragile"}] |
| RETURNS | HTTP code 200 - at successful insert in DB,  HTTP code 500 - at fail. |
| ERRORS | 400 – Not valid parameters (JSON)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/Restrictions/id** | |
| DESCRIPTION | Deletes one exact entity of Restrictions in DB, according availability for logged user role |
| URL STRUCTURE | [/api/Restrictions/id](http://localhost/EFlocal/api/Restrictions/id) |
| METHOD | DELETE |
| PARAMETERS | [FromURI]  name: Id, type: int. Represents Id of targeted entity  [FromBody]  - |
| RETURNS | HTTP code 200 - at successful insert in DB,  HTTP code 500 - at fail. |
| ERRORS | 400 – Not valid parameters (Id, JSON)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/Restrictions** | |
| DESCRIPTION | Updates one exact entity of Restrictions in DB, according availability for logged user role |
| URL STRUCTURE | [/api/Restrictions](http://localhost/EFlocal/api/Restrictions) |
| METHOD | PUT |
| PARAMETERS | [FromURI]  -  [FromBody]  name: restrictions, type: JSON-encoded <RestrictionsDTO>. One entity of updated entity. Sample: same as /api/Restrictions (POST). |
| RETURNS | HTTP code 200 - at successful insert in DB,  HTTP code 500 - at fail. |
| ERRORS | 400 – Not valid parameters (JSON)  401 – User is not authorized  500 – Error at work with DB |

**RestrictionsSet**

RestrictionsSet is one of the supporting entities in application – its API controller represents CRUD functionality for set of restrictions of exact good.

|  |  |
| --- | --- |
| **/api/RestrictionsSet** | |
| DESCRIPTION | Gets the list of all RestrictionsSet, according availability for logged user role |
| URL STRUCTURE | /api/RestrictionsSet |
| METHOD | GET |
| PARAMETERS | API supports OData client-side paging. First of all, works orderby & direction. Skip&top – at the end.  [FromURI]  name: orderby, type: String?. Sets the column sort order for query.  name: direction, type: String?. Sets the direction sort for query. Default: asc. Available: desc.  name: skip, type: int?. Comes in pair with top. Sets number of skipped elements in query for return.  name: top, type: int?. Sets number of displayed elements in query for return.  [FromBody]  - |
| RETURNS | type: JSON-encoded List<RestrictionsSet>. Returns list of all requested entities, available to this User. Sample response: [{"Goods":{"Sizes":{"Id":1,"Size":"40\*60\*20"},"TypeOfStorage":{"Id":1,"Type":"Palette"},"Id":1,"Name":"Salt","Quantity":10,"TypeOfStorageId":1,"SizesId":1,"Info":"1 good info"},"Restrictions":{"Id":1,"RestrictionName":"Gragile"}, "RestrictionsId":1,"GoodsId":1,"Id":1},{"Goods":{"Sizes":{"Id":1,"Size":"40\*60\*20"},"TypeOfStorage":{"Id":1,"Type":"Palette"},"Id":2,"Name":"Pepper","Quantity":5,"TypeOfStorageId":1,"SizesId":1,"Info":"2 info"},"Restrictions":{"Id":2, "RestrictionName":"Flameable"},"RestrictionsId":2,"GoodsId":2,"Id":2}] |
| ERRORS | 400 – Not valid parameters (OData, JSON)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/RestrictionsSet/id** | |
| DESCRIPTION | Gets one exact entity of RestrictionsSet, according availability for logged user role |
| URL STRUCTURE | /api/RestrictionsSet/id |
| METHOD | GET |
| PARAMETERS | [FromURI]  name: Id, type: int. Represents Id of targeted entity  [FromBody]  - |
| RETURNS | type: JSON. One entity if it`s available to this User. Sample response: [{"Goods":{"Sizes":{"Id":1,"Size":"40\*60\*20"},"TypeOfStorage":{"Id":1,"Type":"Palette"},"Id":1,"Name":"Salt","Quantity":10,"TypeOfStorageId":1,"SizesId":1,"Info":"1 good info"},"Restrictions":{"Id":1,"RestrictionName":"Gragile"}, "RestrictionsId":1,"GoodsId":1,"Id":1}] |
| ERRORS | 400 – Not valid parameters (Id, OData)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/RestrictionsSet/All** | |
| DESCRIPTION | This action is available only for administrators. Gets full list of RestrictionsSet. |
| URL STRUCTURE | /api/RestrictionsSet/All |
| METHOD | GET` |
| PARAMETERS | [FromURI]  name: orderby, type: String?. Sets the column sort order for query.  name: direction, type: String?. Sets the direction sort for query. Default: asc. Available: desc.  name: skip, type: int?. Comes in pair with top. Sets number of skipped elements in query for return.  name: top, type: int?. Sets number of displayed elements in query for return.  [FromBody]  - |
| RETURNS | type: JSON. List of RestrictionsSet. Sample of response – same as : /api/RestrictionsSet (GET). |
| ERRORS | 400 – Not valid parameters (OData,)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/RestrictionsSet** | |
| DESCRIPTION | Sends new entity of RestrictionsSet to DB, according availability for logged user role |
| URL STRUCTURE | /api/RestrictionsSet |
| METHOD | POST |
| PARAMETERS | name: restrictionsSet, type: JSON-encoded <RestrictionsSetDTO>. One requested entity. Sample["RestrictionsId":1,"GoodsId":1,"Id":1}] |
| RETURNS | HTTP code 200 - at successful insert in DB,  HTTP code 500 - at fail. |
| ERRORS | 400 – Not valid parameters (JSON)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/RestrictionsSet/id** | |
| DESCRIPTION | Deletes one exact entity of RestrictionsSet in DB, according availability for logged user role |
| URL STRUCTURE | [/api/RestrictionsSet/id](http://localhost/EFlocal/api/RestrictionsSet/id) |
| METHOD | DELETE |
| PARAMETERS | [FromURI]  name: Id, type: int. Represents Id of targeted entity  [FromBody]  - |
| RETURNS | HTTP code 200 - at successful insert in DB,  HTTP code 500 - at fail. |
| ERRORS | 400 – Not valid parameters (Id, JSON)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/RestrictionsSet** | |
| DESCRIPTION | Updates one exact entity of RestrictionsSet in DB, according availability for logged user role |
| URL STRUCTURE | [/api/RestrictionsSet](http://localhost/EFlocal/api/RestrictionsSet) |
| METHOD | PUT |
| PARAMETERS | [FromURI]  -  [FromBody]  name: restrictionsSet, type: JSON-encoded <RestrictionsSetDTO>. One entity of updated entity. Sample: same as /api/RestrictionsSet (POST). |
| RETURNS | HTTP code 200 - at successful insert in DB,  HTTP code 500 - at fail. |
| ERRORS | 400 – Not valid parameters (JSON)  401 – User is not authorized  500 – Error at work with DB |

**Role**

Roles is one of the supporting entities in application – its API controller represents CRUD functionality for managing roles dictionary. Controller enabled only for Administrators.

|  |  |
| --- | --- |
| **/api/Role** | |
| DESCRIPTION | Gets the list of all Roles, according availability for logged user role |
| URL STRUCTURE | /api/Role |
| METHOD | GET |
| PARAMETERS | API supports OData client-side paging. First of all, works orderby & direction. Skip&top – at the end.  [FromURI]  name: orderby, type: String?. Sets the column sort order for query.  name: direction, type: String?. Sets the direction sort for query. Default: asc. Available: desc.  name: skip, type: int?. Comes in pair with top. Sets number of skipped elements in query for return.  name: top, type: int?. Sets number of displayed elements in query for return.  [FromBody]  - |
| RETURNS | type: JSON-encoded List<Role>. Returns list of all requested entities, available to this User. Sample response: [{"Users":[],"Id":1,"Name":"Admin"},{"Users":[],"Id":2,"Name":"Manager"}] |
| ERRORS | 400 – Not valid parameters (OData, JSON)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/Role/id** | |
| DESCRIPTION | Gets one exact entity of Role, according availability for logged user role |
| URL STRUCTURE | /api/Role/id |
| METHOD | GET |
| PARAMETERS | [FromURI]  name: Id, type: int. Represents Id of targeted entity  [FromBody]  - |
| RETURNS | type: JSON. One entity if it`s available to this User. Sample response: [{"Users":[],"Id":1,"Name":"Admin"}] |
| ERRORS | 400 – Not valid parameters (Id, OData)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/Role/All** | |
| DESCRIPTION | This action is available only for administrators. Gets full list of Role. |
| URL STRUCTURE | /api/Role/All |
| METHOD | GET` |
| PARAMETERS | [FromURI]  name: orderby, type: String?. Sets the column sort order for query.  name: direction, type: String?. Sets the direction sort for query. Default: asc. Available: desc.  name: skip, type: int?. Comes in pair with top. Sets number of skipped elements in query for return.  name: top, type: int?. Sets number of displayed elements in query for return.  [FromBody]  - |
| RETURNS | type: JSON. List of Role. Sample of response – same as : /api/Role (GET). |
| ERRORS | 400 – Not valid parameters (OData,)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/Role** | |
| DESCRIPTION | Sends new entity of Role to DB, according availability for logged user role |
| URL STRUCTURE | /api/Role |
| METHOD | POST |
| PARAMETERS | [FromURI]  -  [FromBody]  name: role, type: JSON-encoded <RoleDTO>. One requested entity. Sample: [{"Id":1,"Name":"Admin"}] |
| RETURNS | HTTP code 200 - at successful insert in DB,  HTTP code 500 - at fail. |
| ERRORS | 400 – Not valid parameters (JSON)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/Role/id** | |
| DESCRIPTION | Deletes one exact entity of Role in DB, according availability for logged user role |
| URL STRUCTURE | [/api/Role/id](http://localhost/EFlocal/api/Role/id) |
| METHOD | DELETE |
| PARAMETERS | [FromURI]  name: Id, type: int. Represents Id of targeted entity  [FromBody]  - |
| RETURNS | HTTP code 200 - at successful insert in DB,  HTTP code 500 - at fail. |
| ERRORS | 400 – Not valid parameters (Id, JSON)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/Role** | |
| DESCRIPTION | Updates one exact entity of Role in DB, according availability for logged user role |
| URL STRUCTURE | [/api/Role](http://localhost/EFlocal/api/Roles) |
| METHOD | PUT |
| PARAMETERS | [FromURI]  -  [FromBody]  name: role, type: JSON-encoded <RoleDTO>. One entity of updated entity. Sample: same as /api/Role (POST). |
| RETURNS | HTTP code 200 - at successful insert in DB,  HTTP code 500 - at fail. |
| ERRORS | 400 – Not valid parameters (JSON)  401 – User is not authorized  500 – Error at work with DB |

**Sizes**

Sizes of Goods is one of the supporting entities in application – its API controller represents CRUD functionality for work with its dictionary.

|  |  |
| --- | --- |
| **/api/Sizes** | |
| DESCRIPTION | Gets the list of all Sizes, according availability for logged user role |
| URL STRUCTURE | /api/Sizes |
| METHOD | GET |
| PARAMETERS | API supports OData client-side paging. First of all, works orderby & direction. Skip&top – at the end.  [FromURI]  name: orderby, type: String?. Sets the column sort order for query.  name: direction, type: String?. Sets the direction sort for query. Default: asc. Available: desc.  name: skip, type: int?. Comes in pair with top. Sets number of skipped elements in query for return.  name: top, type: int?. Sets number of displayed elements in query for return.  [FromBody]  - |
| RETURNS | type: JSON-encoded List<Sizes>. Returns list of all requested entities, available to this User. Sample response: [{"Id":1,"Size":"40\*60\*20"}, {"Id":2,"Size":"40\*40\*40"}] |
| ERRORS | 400 – Not valid parameters (OData, JSON)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/Sizes/id** | |
| DESCRIPTION | Gets one exact entity of Sizes, according availability for logged user role |
| URL STRUCTURE | /api/Sizes/id |
| METHOD | GET |
| PARAMETERS | [FromURI]  name: Id, type: int. Represents Id of targeted entity  [FromBody]  - |
| RETURNS | type: JSON. One entity if it`s available to this User. Sample response: [{"Id":1,"Size":"40\*60\*20"}] |
| ERRORS | 400 – Not valid parameters (Id, OData)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/Sizes/All** | |
| DESCRIPTION | This action is available only for administrators. Gets full list of Transactions. |
| URL STRUCTURE | /api/Sizes/All |
| METHOD | GET |
| PARAMETERS | [FromURI]  name: orderby, type: String?. Sets the column sort order for query.  name: direction, type: String?. Sets the direction sort for query. Default: asc. Available: desc.  name: skip, type: int?. Comes in pair with top. Sets number of skipped elements in query for return.  name: top, type: int?. Sets number of displayed elements in query for return.  [FromBody]  - |
| RETURNS | type: JSON. List of Transactions. Sample of response – same as : /api/Sizes (GET). |
| ERRORS | 400 – Not valid parameters (OData,)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/Sizes** | |
| DESCRIPTION | Sends new entity of Sizes to DB, according availability for logged user role |
| URL STRUCTURE | /api/Sizes |
| METHOD | POST |
| PARAMETERS | [FromURI]  -  [FromBody]  name: sizes, type: JSON-encoded <SizesDTO>. One requested entity. Sample: [{"Id":1,"Size":"40\*60\*20"}] |
| RETURNS | HTTP code 200 - at successful insert in DB,  HTTP code 500 - at fail. |
| ERRORS | 400 – Not valid parameters (JSON)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/Sizes/id** | |
| DESCRIPTION | Deletes one exact entity of Sizes in DB, according availability for logged user role |
| URL STRUCTURE | [/api/Sizes/id](http://localhost/EFlocal/api/Sizes/id) |
| METHOD | DELETE |
| PARAMETERS | [FromURI]  name: Id, type: int. Represents Id of targeted entity  [FromBody]  - |
| RETURNS | HTTP code 200 - at successful insert in DB,  HTTP code 500 - at fail. |
| ERRORS | 400 – Not valid parameters (Id, JSON)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/Sizes** | |
| DESCRIPTION | Updates one exact entity of Sizes in DB, according availability for logged user role |
| URL STRUCTURE | [/api/Sizes](http://localhost/EFlocal/api/Sizes) |
| METHOD | PUT |
| PARAMETERS | [FromURI]  -  [FromBody]  name: sizes, type: JSON-encoded <SizesDTO>. One entity of updated entity. Sample: same as /api/Sizes (POST). |
| RETURNS | HTTP code 200 - at successful insert in DB,  HTTP code 500 - at fail. |
| ERRORS | 400 – Not valid parameters (JSON)  401 – User is not authorized  500 – Error at work with DB |

**TranactionType**

TranactionType is one of the supporting entities in application – its API controller represents CRUD functionality for managing dictionary of transaction types.

|  |  |
| --- | --- |
| **/api/TranactionType** | |
| DESCRIPTION | Gets the list of all TranactionType, according availability for logged user role |
| URL STRUCTURE | /api/TranactionType |
| METHOD | GET |
| PARAMETERS | API supports OData client-side paging. First of all, works orderby & direction. Skip&top – at the end.  [FromURI]  name: orderby, type: String?. Sets the column sort order for query.  name: direction, type: String?. Sets the direction sort for query. Default: asc. Available: desc.  name: skip, type: int?. Comes in pair with top. Sets number of skipped elements in query for return.  name: top, type: int?. Sets number of displayed elements in query for return.  [FromBody]  - |
| RETURNS | type: JSON-encoded List<TranactionType>. Returns list of all requested entities, available to this User. Sample response: [{"Id":1,"Name":"Supply"}, {"Id":2,"Name":"Order"}] |
| ERRORS | 400 – Not valid parameters (OData, JSON)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/TranactionType/id** | |
| DESCRIPTION | Gets one exact entity of TranactionType, according availability for logged user role |
| URL STRUCTURE | /api/TranactionType/id |
| METHOD | GET |
| PARAMETERS | [FromURI]  name: Id, type: int. Represents Id of targeted entity  [FromBody]  - |
| RETURNS | type: JSON. One entity if it`s available to this User. Sample response: [{"Id":1,"Name":"Supply"}] |
| ERRORS | 400 – Not valid parameters (Id, OData)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/TranactionType/All** | |
| DESCRIPTION | This action is available only for administrators. Gets full list of TranactionType. |
| URL STRUCTURE | /api/TranactionType/All |
| METHOD | GET |
| PARAMETERS | [FromURI]  name: orderby, type: String?. Sets the column sort order for query.  name: direction, type: String?. Sets the direction sort for query. Default: asc. Available: desc.  name: skip, type: int?. Comes in pair with top. Sets number of skipped elements in query for return.  name: top, type: int?. Sets number of displayed elements in query for return.  [FromBody]  - |
| RETURNS | type: JSON. List of TranactionType. Sample of response – same as : /api/TranactionType (GET). |
| ERRORS | 400 – Not valid parameters (OData,)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/TranactionType** | |
| DESCRIPTION | Sends new entity of TranactionType to DB, according availability for logged user role |
| URL STRUCTURE | /api/TranactionType |
| METHOD | POST |
| PARAMETERS | name: tranactionType, type: JSON-encoded <TransactionTypeDTO>. One requested entity. Sample: [{"Id":1,"Name":"Supply"}]. |
| RETURNS | HTTP code 200 - at successful insert in DB,  HTTP code 500 - at fail. |
| ERRORS | 400 – Not valid parameters (JSON)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/TranactionType/id** | |
| DESCRIPTION | Deletes one exact entity of TranactionType in DB, according availability for logged user role |
| URL STRUCTURE | [/api/TranactionType/id](http://localhost/EFlocal/api/TranactionType/id) |
| METHOD | DELETE |
| PARAMETERS | [FromURI]  name: Id, type: int. Represents Id of targeted entity  [FromBody]  - |
| RETURNS | HTTP code 200 - at successful insert in DB,  HTTP code 500 - at fail. |
| ERRORS | 400 – Not valid parameters (Id, JSON)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/TranactionType** | |
| DESCRIPTION | Updates one exact entity of TranactionType in DB, according availability for logged user role |
| URL STRUCTURE | [/api/TranactionType](http://localhost/EFlocal/api/TranactionType) |
| METHOD | PUT |
| PARAMETERS | [FromURI]  -  [FromBody]  name: transaction, type: JSON-encoded <TransactionTypeDTO>. One entity of updated entity. Sample: same as /api/TransactionType (POST). |
| RETURNS | HTTP code 200 - at successful insert in DB,  HTTP code 500 - at fail. |
| ERRORS | 400 – Not valid parameters (JSON)  401 – User is not authorized  500 – Error at work with DB |

**Transactions**

Transactions is one of the main entities in application – its API controller represents CRUD functionality for goods moving activity.

|  |  |
| --- | --- |
| **/api/Transactions** | |
| DESCRIPTION | Gets the list of current Users transactions |
| URL STRUCTURE | /api/Transactions |
| METHOD | GET |
| PARAMETERS | API supports OData client-side paging. First of all, works orderby & direction. Skip&top – at the end.  [FromURI]  name: orderby, type: String?. Sets the column sort order for query.  name: direction, type: String?. Sets the direction sort for query. Default: asc. Available: desc.  name: skip, type: int?. Comes in pair with top. Sets number of skipped elements in query for return.  name: top, type: int?. Sets number of displayed elements in query for return.  [FromBody]  - |
| RETURNS | type: JSON-encoded List<Transactions>. Returns list of all requested entities, available to this User. Sample response: [{"TranactionType":{"Id":2,"Name":"Order"},"Users":{"Email":"1@2.com","UserName":"1@2.com","Id":1},"Description":"555", "TranactionTypeId":2, "UserId":1,"Date":"2016-06-05T19:37:30","Id":10023}, {"TranactionType":{"Id":1, "Name":"Supply"}, "Users":{"Email":"1@2.com", "UserName":"1@2.com", "Id":1}, "Description":"666", "TranactionTypeId":1, "UserId":1,"Date":"2016-06-03T19:01:44","Id":10017}] |
| ERRORS | 400 – Not valid parameters (OData, JSON)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/Transactions/id** | |
| DESCRIPTION | Gets one exact entity of Transaction of logged User |
| URL STRUCTURE | [/api/Transactions/id](http://localhost/EFlocal/api/Transactions/id) |
| METHOD | GET |
| PARAMETERS | [FromURI]  name: Id, type: int. Represents Id of targeted entity  [FromBody]  - |
| RETURNS | type: JSON. One entity if it`s available to this User. Sample response: [{"TranactionType":{"Id":2,"Name":"Order"},"Users":{"Email":"1@2.com","UserName":"1@2.com","Id":1},"Description":"555 & ViewBags & DropDownList", "TranactionTypeId":2, "UserId":1,"Date":"2016-06-05T19:37:30","Id":10023}] |
| ERRORS | 400 – Not valid parameters (Id, OData, JSON)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/Transactions/All** | |
| DESCRIPTION | This action is available only for administrators. Gets full list of Transactions. |
| URL STRUCTURE | /api/Transactions/All |
| METHOD | GET` |
| PARAMETERS | [FromURI]  name: orderby, type: String?. Sets the column sort order for query.  name: direction, type: String?. Sets the direction sort for query. Default: asc. Available: desc.  name: skip, type: int?. Comes in pair with top. Sets number of skipped elements in query for return.  name: top, type: int?. Sets number of displayed elements in query for return.  [FromBody]  - |
| RETURNS | type: JSON. List of Transactions. Sample of response – same as : /api/Transactions (GET). |
| ERRORS | 400 – Not valid parameters (OData,)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/Transactions** | |
| DESCRIPTION | Sends new entity of Transactions to DB, according availability for logged user role |
| URL STRUCTURE | /api/Transactions |
| METHOD | POST |
| PARAMETERS | [FromURI]  -  [FromBody]  name: transaction, type: JSON-encoded <TransactionDTO>. One requested entity. Sample: [{"Description":"555", "TranactionTypeId":2, "UserId":1, "Date":"2016-06-05T19:37:30", "Id":10023}] |
| RETURNS | HTTP code 200 - at successful insert in DB,  HTTP code 500 - at fail. |
| ERRORS | 400 – Not valid parameters (JSON)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/Transactions/id** | |
| DESCRIPTION | Deletes one exact entity of Transactions in DB, according availability for logged user role |
| URL STRUCTURE | [/api/Transactions/id](http://localhost/EFlocal/api/Transactions/id) |
| METHOD | DELETE |
| PARAMETERS | [FromURI]  name: Id, type: int. Represents Id of targeted entity  [FromBody]  - |
| RETURNS | HTTP code 200 - at successful insert in DB,  HTTP code 500 - at fail. |
| ERRORS | 400 – Not valid parameters (Id, JSON)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/Transactions** | |
| DESCRIPTION | Updates one exact entity of Transactions in DB, according availability for logged user role |
| URL STRUCTURE | [/api/Transactions](http://localhost/EFlocal/api/Transactions) |
| METHOD | PUT |
| PARAMETERS | [FromURI]  -  [FromBody]  name: transaction, type: JSON-encoded TransactionDTO. One entity of updated entity. Sample: same as /api/Transactions (POST). |
| RETURNS | HTTP code 200 - at successful insert in DB,  HTTP code 500 - at fail. |
| ERRORS | 400 – Not valid parameters (JSON)  401 – User is not authorized  500 – Error at work with DB |

**TypeOfStorage**

TypeOfStorage is one of the supporting entities in application – its API controller represents CRUD functionality for managing its dictionary.

|  |  |
| --- | --- |
| **/api/TypeOfStorage** | |
| DESCRIPTION | Gets the list of all TypeOfStorage, according availability for logged user role |
| URL STRUCTURE | /api/TypeOfStorage |
| METHOD | GET |
| PARAMETERS | API supports OData client-side paging. First of all, works orderby & direction. Skip&top – at the end.  [FromURI]  name: orderby, type: String?. Sets the column sort order for query.  name: direction, type: String?. Sets the direction sort for query. Default: asc. Available: desc.  name: skip, type: int?. Comes in pair with top. Sets number of skipped elements in query for return.  name: top, type: int?. Sets number of displayed elements in query for return.  [FromBody]  - |
| RETURNS | type: JSON-encoded List<TypeOfStorage>. Returns list of all requested entities, available to this User. Sample response: [{"Id":1,"Type":"Palette"},{"Id":2,"Type":"Pile"}] |
| ERRORS | 400 – Not valid parameters (OData, JSON)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/TypeOfStorage/id** | |
| DESCRIPTION | Gets one exact entity of TypeOfStorage, according availability for logged user role |
| URL STRUCTURE | /api/TypeOfStorage/id |
| METHOD | GET |
| PARAMETERS | [FromURI]  name: Id, type: int. Represents Id of targeted entity  [FromBody]  - |
| RETURNS | type: JSON. One entity if it`s available to this User. Sample response: [{"Id":1,"Type":"Palette"}] |
| ERRORS | 400 – Not valid parameters (Id, OData)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/TypeOfStorage/All** | |
| DESCRIPTION | This action is available only for administrators. Gets full list of TypeOfStorage. |
| URL STRUCTURE | /api/TypeOfStorage/All |
| METHOD | GET |
| PARAMETERS | [FromURI]  name: orderby, type: String?. Sets the column sort order for query.  name: direction, type: String?. Sets the direction sort for query. Default: asc. Available: desc.  name: skip, type: int?. Comes in pair with top. Sets number of skipped elements in query for return.  name: top, type: int?. Sets number of displayed elements in query for return.  [FromBody]  - |
| RETURNS | type: JSON. List of Transactions. Sample of response – same as : /api/TypeOfStorage (GET). |
| ERRORS | 400 – Not valid parameters (OData,)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/TypeOfStorage** | |
| DESCRIPTION | Sends new entity of TypeOfStorage to DB, according availability for logged user role |
| URL STRUCTURE | /api/TypeOfStorage |
| METHOD | POST |
| PARAMETERS | [FromURI]  -  [FromBody]  name: transaction, type: JSON-encoded <TypeOfStorageDTO>. One requested entity. Sample: [{"Id":1,"Type":"Palette"}] |
| RETURNS | HTTP code 200 - at successful insert in DB,  HTTP code 500 - at fail. |
| ERRORS | 400 – Not valid parameters (JSON)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/TypeOfStorage/id** | |
| DESCRIPTION | Deletes one exact entity of TypeOfStorage in DB, according availability for logged user role |
| URL STRUCTURE | [/api/TypeOfStorage/id](http://localhost/EFlocal/api/TypeOfStorage/id) |
| METHOD | DELETE |
| PARAMETERS | [FromURI]  name: Id, type: int. Represents Id of targeted entity  [FromBody]  - |
| RETURNS | HTTP code 200 - at successful insert in DB,  HTTP code 500 - at fail. |
| ERRORS | 400 – Not valid parameters (Id, JSON)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/TypeOfStorage** | |
| DESCRIPTION | Updates one exact entity of TypeOfStorage in DB, according availability for logged user role |
| URL STRUCTURE | [/api/TypeOfStorage](http://localhost/EFlocal/api/TypeOfStorage) |
| METHOD | PUT |
| PARAMETERS | [FromURI]  -  [FromBody]  name: transaction, type: JSON-encoded <TypeOfStorageDTO>. One entity of updated entity. Sample: same as /api/ TypeOfStorage (POST). |
| RETURNS | HTTP code 200 - at successful insert in DB,  HTTP code 500 - at fail. |
| ERRORS | 400 – Not valid parameters (JSON)  401 – User is not authorized  500 – Error at work with DB |

**User**

User is one of the main entities in application – its API controller represents CRUD functionality for managing its dictionary. Controller enabled only for Administrators.

|  |  |
| --- | --- |
| **/api/User** | |
| DESCRIPTION | Gets the list of all User. |
| URL STRUCTURE | /api/User |
| METHOD | GET |
| PARAMETERS | API supports OData client-side paging. First of all, works orderby & direction. Skip&top – at the end.  [FromURI]  name: orderby, type: String?. Sets the column sort order for query.  name: direction, type: String?. Sets the direction sort for query. Default: asc. Available: desc.  name: skip, type: int?. Comes in pair with top. Sets number of skipped elements in query for return.  name: top, type: int?. Sets number of displayed elements in query for return.  [FromBody]  - |
| RETURNS | type: JSON-encoded List<User>. Returns list of all requested entities. Sample response: [{"Email":"2@2.com","UserName":"2@2.com","Id":2},{"Email":"3@2.com","UserName":"3@2.com","Id":10003}] |
| ERRORS | 400 – Not valid parameters (OData, JSON)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/User/id** | |
| DESCRIPTION | Gets one exact entity of User. |
| URL STRUCTURE | /api/User/id |
| METHOD | GET |
| PARAMETERS | [FromURI]  name: Id, type: int. Represents Id of targeted entity  [FromBody]  - |
| RETURNS | type: JSON. One of User. Sample response: [{"Email":"2@2.com","UserName":"2@2.com","Id":2}] |
| ERRORS | 400 – Not valid parameters (Id, OData)  401 – User is not authorized  500 – Error at work with DB |
| **/api/User** | |
| DESCRIPTION | Sends new entity of User to DB. |
| URL STRUCTURE | /api/User |
| METHOD | POST |
| PARAMETERS | name: user, type: JSON-encoded <UserDTO>. One requested entity. Sample: [{"TwoFactorEnabled":true, "SecurityStamp":"akodOI3\*", "PhoneNumberConfirmed":false, "PhoneNumber":"555-55-55", "PasswordHash":"ASDojdsi#", "LockoutEndDateUtc":"2016-06-05T19:37:30", "LockoutEnabled":false, "EmailConfirmed":false, "AccessFailedCount":2, "Email":"2@2.com", "UserName":"2@2.com", "Id":2}] |
| RETURNS | HTTP code 200 - at successful insert in DB,  HTTP code 500 - at fail. |
| ERRORS | 400 – Not valid parameters (JSON)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/User/id** | |
| DESCRIPTION | Deletes one exact entity of User in DB. |
| URL STRUCTURE | [/api/User/id](http://localhost/EFlocal/api/User/id) |
| METHOD | DELETE |
| PARAMETERS | [FromURI]  name: Id, type: int. Represents Id of targeted entity  [FromBody]  - |
| RETURNS | HTTP code 200 - at successful insert in DB,  HTTP code 500 - at fail. |
| ERRORS | 400 – Not valid parameters (Id, JSON)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/User** | |
| DESCRIPTION | Updates one exact entity of User in DB |
| URL STRUCTURE | [/api/User](http://localhost/EFlocal/api/Users) |
| METHOD | PUT |
| PARAMETERS | [FromURI]  -  [FromBody]  name: user, type: JSON-encoded <UserDTO>. One entity of updated entity. Sample: same as /api/User (POST). |
| RETURNS | HTTP code 200 - at successful insert in DB,  HTTP code 500 - at fail. |
| ERRORS | 400 – Not valid parameters (JSON)  401 – User is not authorized  500 – Error at work with DB |

**UserRole**

UserRole is one of the supporting entities in application – its API controller represents CRUD functionality for managing its dictionary. Controller enabled only for Administrators.

|  |  |
| --- | --- |
| **/api/UserRole** | |
| DESCRIPTION | Gets the list of all UserRole. |
| URL STRUCTURE | /api/UserRole |
| METHOD | GET |
| PARAMETERS | API supports OData client-side paging. First of all, works orderby & direction. Skip&top – at the end.  [FromURI]  name: orderby, type: String?. Sets the column sort order for query.  name: direction, type: String?. Sets the direction sort for query. Default: asc. Available: desc.  name: skip, type: int?. Comes in pair with top. Sets number of skipped elements in query for return.  name: top, type: int?. Sets number of displayed elements in query for return.  [FromBody]  - |
| RETURNS | type: JSON-encoded List<UserRole>. Returns list of all requested entities. Sample response: [{"Id":2,"RoleId":2,"UserId":2}, {"Id":10003,"RoleId":3,"UserId":10003}] |
| ERRORS | 400 – Not valid parameters (OData, JSON)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/UserRole/id** | |
| DESCRIPTION | Gets one exact entity of UserRole |
| URL STRUCTURE | /api/UserRole/id |
| METHOD | GET |
| PARAMETERS | [FromURI]  name: Id, type: int. Represents Id of targeted entity  [FromBody]  - |
| RETURNS | type: JSON. One entity. Sample response: [{"Id":2,"RoleId":2,"UserId":2}] |
| ERRORS | 400 – Not valid parameters (Id, OData)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/UserRole** | |
| DESCRIPTION | Sends new entity of UserRole to DB |
| URL STRUCTURE | /api/UserRole |
| METHOD | POST |
| PARAMETERS | [FromURI]  -  [FromBody]  name: userRole, type: JSON-encoded <UserRoleDTO>. One requested entity. Sample: [{"Id":2,"RoleId":2,"UserId":2}] |
| RETURNS | HTTP code 200 - at successful insert in DB,  HTTP code 500 - at fail. |
| ERRORS | 400 – Not valid parameters (JSON)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/UserRole/id** | |
| DESCRIPTION | Deletes one exact entity of UserRole in DB. |
| URL STRUCTURE | [/api/UserRole/id](http://localhost/EFlocal/api/UserRole/id) |
| METHOD | DELETE |
| PARAMETERS | [FromURI]  name: Id, type: int. Represents Id of targeted entity  [FromBody]  - |
| RETURNS | HTTP code 200 - at successful insert in DB,  HTTP code 500 - at fail. |
| ERRORS | 400 – Not valid parameters (Id, JSON)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/UserRole** | |
| DESCRIPTION | Updates one exact entity of UserRole in DB |
| URL STRUCTURE | [/api/UserRole](http://localhost/EFlocal/api/UserRole) |
| METHOD | PUT |
| PARAMETERS | [FromURI]  -  [FromBody]  name: userRole, type: JSON-encoded <UserRoleDTO>. One entity of updated entity. Sample: same as /api/UserRole (POST). |
| RETURNS | HTTP code 200 - at successful insert in DB,  HTTP code 500 - at fail. |
| ERRORS | 400 – Not valid parameters (JSON)  401 – User is not authorized  500 – Error at work with DB |

**WarehousesPlaces**

WarehousesPlaces is one of the supporting entities in application – its API controller represents CRUD functionality for managing its dictionary.

|  |  |
| --- | --- |
| **/api/WarehousesPlaces** | |
| DESCRIPTION | Gets the list of all WarehousesPlaces, according availability for logged user role |
| URL STRUCTURE | /api/WarehousesPlaces |
| METHOD | GET |
| PARAMETERS | API supports OData client-side paging. First of all, works orderby & direction. Skip&top – at the end.  [FromURI]  name: orderby, type: String?. Sets the column sort order for query.  name: direction, type: String?. Sets the direction sort for query. Default: asc. Available: desc.  name: skip, type: int?. Comes in pair with top. Sets number of skipped elements in query for return.  name: top, type: int?. Sets number of displayed elements in query for return.  [FromBody]  - |
| RETURNS | type: JSON-encoded List<WarehousesPlaces>. Returns list of all requested entities, available to this User. Sample response: [{"Id":1,"Adress":"Kharkiv, first WH","Place":"1"},{"Id":2,"Adress":"Kharkiv, first WH","Place":"1A"}] |
| ERRORS | 400 – Not valid parameters (OData, JSON)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/WarehousesPlaces/id** | |
| DESCRIPTION | Gets one exact entity of WarehousesPlaces, according availability for logged user role |
| URL STRUCTURE | /api/WarehousesPlaces/id |
| METHOD | GET |
| PARAMETERS | [FromURI]  name: Id, type: int. Represents Id of targeted entity  [FromBody]  - |
| RETURNS | type: JSON. One entity if it`s available to this User. Sample response: [{"Id":1,"Adress":"Kharkiv, first WH","Place":"1"}] |
| ERRORS | 400 – Not valid parameters (Id, OData)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/WarehousesPlaces/All** | |
| DESCRIPTION | This action is available only for administrators. Gets full list of WarehousesPlaces. |
| URL STRUCTURE | /api/WarehousesPlaces/All |
| METHOD | GET |
| PARAMETERS | [FromURI]  name: orderby, type: String?. Sets the column sort order for query.  name: direction, type: String?. Sets the direction sort for query. Default: asc. Available: desc.  name: skip, type: int?. Comes in pair with top. Sets number of skipped elements in query for return.  name: top, type: int?. Sets number of displayed elements in query for return.  [FromBody]  - |
| RETURNS | type: JSON. List of WarehousesPlaces. Sample of response – same as : /api/WarehousesPlaces (GET). |
| ERRORS | 400 – Not valid parameters (OData,)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/WarehousesPlaces** | |
| DESCRIPTION | Sends new entity of WarehousesPlaces to DB, according availability for logged user role |
| URL STRUCTURE | /api/WarehousesPlaces |
| METHOD | POST |
| PARAMETERS | [FromURI]  -  [FromBody]  name: warehousesPlaces type: JSON-encoded < WarehousesPlacesDTO>. One requested entity. Sample: [{"Id":1,"Adress":"Kharkiv, first WH","Place":"1"}] |
| RETURNS | HTTP code 200 - at successful insert in DB,  HTTP code 500 - at fail. |
| ERRORS | 400 – Not valid parameters (JSON)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/WarehousesPlaces/id** | |
| DESCRIPTION | Deletes one exact entity of WarehousesPlaces in DB, according availability for logged user role |
| URL STRUCTURE | [/api/WarehousesPlaces/id](http://localhost/EFlocal/api/WarehousesPlaces/id) |
| METHOD | DELETE |
| PARAMETERS | [FromURI]  name: Id, type: int. Represents Id of targeted entity  [FromBody]  - |
| RETURNS | HTTP code 200 - at successful insert in DB,  HTTP code 500 - at fail. |
| ERRORS | 400 – Not valid parameters (Id, JSON)  401 – User is not authorized  500 – Error at work with DB |

|  |  |
| --- | --- |
| **/api/WarehousesPlaces** | |
| DESCRIPTION | Updates one exact entity of WarehousesPlaces in DB, according availability for logged user role |
| URL STRUCTURE | [/api/WarehousesPlaces](http://localhost/EFlocal/api/WarehousesPlaces/id) |
| METHOD | PUT |
| PARAMETERS | [FromURI]  -  [FromBody]  name: warehousesPlaces, type: JSON-encoded <WarehousesPlacesDTO>. One entity of updated entity. Sample: same as /api/WarehousesPlaces (POST). |
| RETURNS | HTTP code 200 - at successful insert in DB,  HTTP code 500 - at fail. |
| ERRORS | 400 – Not valid parameters (JSON)  401 – User is not authorized  500 – Error at work with DB |