

Credentials Api Documentation

Toci

August 21, 2015

Contents

1 Introduction	2
1.1 Resource destination:	2
1.2 Error Handling:	2
2 Save data for resource operation.	2
2.1 Resource destination:	2
2.2 Request body:	2
2.3 Example JSON request structure:	3
2.4 Response structure description:	3
3 Retrieve data for resource operation.	3
3.1 Resource destination:	3
3.2 Request body:	3
3.3 Example JSON request structure:	3
3.4 Example JSON response structure:	3
3.5 Response structure description:	4
4 Retrieve data list for resource operation.	4
4.1 Resource destination:	4
4.2 Request body:	4
4.3 Example JSON request structure:	4
4.4 Example JSON response structure:	4
4.5 Response structure description:	4
5 Delete data for resource operation.	5
5.1 Resource destination:	5
5.2 Request body:	5
5.3 Example JSON request structure:	5
5.4 Example JSON response structure:	5
5.5 Response structure description:	5
6 Update data for resource operation.	5
6.1 Resource destination:	5
6.2 Request body:	5
6.3 Example JSON request structure:	6
6.4 Example JSON response structure:	6
6.5 Response structure description:	6

1 Introduction

Resource: /models/Project models/Projects

1.1 Resource destination:

This resource allows to Load/Save/Update and Delete projects in database, with required scope. Project data are stored encrypted in the database, so only it can be only read having the right credential (scope).

1.2 Error Handling:

The error handling is unified in the entire API, and the error list is created bitwise. We use following error codes:

- 0 - success
- 1 - IdMissing
- 2 - WrongId
- 4 - DataMissing
- 8 - PasswordMissing
- 16 - NameMissing
- 32 - WrongData

Error message example: Cannot save the project., Project name exist.

2 Save data for resource operation.

Resource: /Save/Project

2.1 Resource destination:

This resource allows to Save project in database, encrypting the project data.

Example usage: POST /Save/Project

2.2 Request body:

- projectName - string, is a name of project
- projectAuthor - string, is a name of author
- projectData - string, is a data, which will be encrypted
- scopeName - string, is a name of scope, which will allow to delete, update, load this project

2.3 Example JSON request structure:

```
{"projectName": "Name",  
"projectAuthor": "Me",  
"projectData": "Data",  
"scopeName": "scope"}
```

Response formats: JSON

Example JSON Response structure:

```
{"code": "0",  
"message": "Saved!"}
```

2.4 Response structure description:

- code - the error code specified in the "Error Handling" section
- message - response message from api

Remark: All body parameters required.

3 Retrieve data for resource operation.

Resource: /Load/Project

3.1 Resource destination:

This resource allows to load project from database, decrypting the project data, for projectName and required scope.

Example usage: POST /Load/Project

3.2 Request body:

- projectName - string, is a name of project
- scopesList - string, list of strings separated by decimal delete, update, load this project

3.3 Example JSON request structure:

```
{"projectName": "Name",  
"scopesList": "scope, scope2"}
```

3.4 Example JSON response structure:

```
{"code": "0",  
"message": "Loaded!",  
"data": {"Result":  
{"scopeName": "scope",  
"projectId": "8",  
"projectName": "Name",  
"projectData": "Data",  
"projectAuthor": "Me",  
"modificationdate": "21.08.2015 09:59:43"}}}}
```

3.5 Response structure description:

- code - the error code specified in the "Error Handling" section
- message - response message from api
- data/result - the requested content

Remark: All body parameters required.

4 Retrieve data list for resource operation.

Resource: /Load/Project

4.1 Resource destination:

This resource allows to load all projects from database, decrypting the project data for required scopes.

Example usage: POST /Load/Project

4.2 Request body:

- scopesList - string, list of strings separated by decimal

4.3 Example JSON request structure:

```
{"scopesList":"scope, scope2"}
```

4.4 Example JSON response structure:

```
{ "code": "0",  
  "message": "Loaded!",  
  "data": { "Result":  
    [ { "scopeName": "scope",  
        "projectId": "8",  
        "projectName": "Name",  
        "projectData": "Data",  
        "projectAuthor": "Me",  
        "modificationdate": "21.08.2015 09:59:43"},  
      { "scopeName": "scope2",  
        "projectId": "10",  
        "projectName": "2Name",  
        "projectData": "Data2",  
        "projectAuthor": "Me2",  
        "modificationdate": "21.08.2015 12:34:53"} ] } }
```

4.5 Response structure description:

- code - the error code specified in the "Error Handling" section
- message - response message from api
- data/result - the requested content

Remark: All body parameters required.

5 Delete data for resource operation.

Resource: /Delete/Project

5.1 Resource destination:

This resource allows to delete project from database, for projectName and required scope.

Example usage: POST /Delete/Project

5.2 Request body:

- projectName - string, is a name of project
- scopesList - string, list of strings separated by decimal

5.3 Example JSON request structure:

```
{"projectName": "Name",  
"scopesList": "scope, scope2"}
```

5.4 Example JSON response structure:

```
{"code": "0",  
"message": "Deleted!"}
```

5.5 Response structure description:

- code - the error code specified in the "Error Handling" section
- message - response message from api

Remark: All body parameters required.

6 Update data for resource operation.

Resource: /Update/Project

6.1 Resource destination:

This resource allows to update project from database, for required projectId and scope.

Example usage: POST /Update/Project

6.2 Request body:

- projectId - int, project Id
- projectName - string, is a name of project
- scopesList - string, list of strings separated by decimal
- projectData - string, is a data, which will be encrypted
- projectAuthor - string, is a name of author
- scopeName - new scope name

6.3 Example JSON request structure:

```
{ "projectId": "10",  
  "scopesList": "scope, scope2",  
  "projectData": "Data2" }
```

6.4 Example JSON response structure:

```
{ "code": "0",  
  "message": "Updated!" }
```

6.5 Response structure description:

- code - the error code specified in the "Error Handling" section
- message - response message from api

Remark: ProjectId and scopesList are required for request body parameters.