CPB US Backend Candidate Lab

API reference

by Juan Manuel Alvarez ([juan.alvarez@fdvsolutions.com](mailto:juan.alvarez@fdvsolutions.com))

Base URL for the demo site: **http://cpb.fdvs.com.ar/ws**

When sending parameters use x**-www-form-urlencoded**

# User related API

## Create user

|  |  |
| --- | --- |
| **URL** | /auth/createAccount |
| **Method** | POST |
| **Params** | username, email, password |

Returns a json object like this:

{

"codigo": "OK",

"mensaje": "Account created",

"userId": 2

}

If it fails, you’ll get an error 500 http response with proper error message.

## Log in

|  |  |
| --- | --- |
| **URL** | /auth/login |
| **Method** | POST |
| **Params** | username, password |

Returns a json object like this:

{

"codigo": "200",

"mensaje": "OK",

"userId": 1

}

The **userId** parameter is your user id, you’ll later need this value to use the API.

If it fails, you’ll get a json response like this

{

"codigo": "500",

"mensaje": "LOGIN INCORRECT",

"userId": null

}

# Notes related API

The API related to notes is considered to be a private API since it will give access to user specific data, so user and password must be provided on every call. For the sake of simplicity in this demo, user and password must be sent as http BASIC AUTH, which is ok if the channel is secure (SSL).

## Create a Note

|  |  |
| --- | --- |
| **URL** | /notes/**<user\_id>**/add |
| **Method** | POST |
| **Params** | title, description |

Returns a json object like this:

{

"code": "OK",

"message": "Note created",

"noteId": 1

}

If it fails, you’ll get an error 500 http response with proper error message.

## Delete a Note

|  |  |
| --- | --- |
| **URL** | /notes/**<user\_id>**/note/**<note\_id>**/delete |
| **Method** | POST |
| **Params** | <none> |

Returns a json object like this:

{

"code": "OK",

"message": "Note deleted"

}

If it fails, you’ll get an error 500 http response with proper error message.

## Update a Note

|  |  |
| --- | --- |
| **URL** | /notes/**<user\_id>**/note/**<note\_id>**/update |
| **Method** | POST |
| **Params** | title, description |

Returns a json object like this:

{

"code": "OK",

"message": "Note updated",

"noteId": 3,

"date": 1376872172992

}

If it fails, you’ll get an error 500 http response with proper error message.

## Get a Note

|  |  |
| --- | --- |
| **URL** | /notes/**<user\_id>**/note/**<note\_id>**/get |
| **Method** | GET |
| **Params** | none |

Returns a json object like this:

{

"id": 2,

"ownerId": 1,

"title": "this is my note",

"description": "this is the text of my note",

"created": 1376871850000,

"lastModified": 1376871850000

}

If it fails, you’ll get an error 500 http response with proper error message.

## List Notes

|  |  |
| --- | --- |
| **URL** | /notes/**<user\_id>**/list |
| **Method** | GET |
| **Params** | Optional:   * **pageSize**: number from 1 or bigger * **pageNumber**: number from 1 or bigger |

Returns a json object like this:

[

{

"id": 2,

"ownerId": 1,

"title": "this is my note",

"description": "my note 1",

"created": 1376871850000,

"lastModified": 1376871850000

},

{

"id": 3,

"ownerId": 1,

"title": "this is my note",

"description": "my note 2",

"created": 1376871851000,

"lastModified": 1376872172000

},

…

]

If it fails, you’ll get an error 500 http response with proper error message.