HOODIE cheat sheet

Initialize Hoodie

var hoodie = new Hoodie()

Init Hoodie on your server

new Hoodie(‘https://example.com‘)

Init Hoodie on different server

hoodie.store

All methods return a promise.

.add(type, properties)

add data

.find(type, id)

find specific data with ID

.findAll(type)

find all data of type

.update(type, id, changedProperties)

update specific data of type with ID

.updateAll(type, changedProperties)

update all data of type

.remove(type, id)

remove specific data with ID

.removeAll(type, id)

remove all data of type

.findOrAdd(type, id, properties)

find data and add if not there

.UpdateOrAdd(type, id, properties)

add data or update, if already there

hoodie.account

All methods return a promise.

.signUp(username, password)

sign user up

.signIn(username, password)

sign user in

.signOut()

sign user out

.destroy()

destroy user account, delete all data

.resetPassword(username)

reset password of user account

.username

gives back string of username

others

hoodie.id()

returns unique ID for current user

hoodie.request(type, path, options)

send custom requests, type being „GET“, „POST“

var rStore = hoodie.open(dbName)

work with remote store, same API as hoodie.store

events - hoodie.on(‘event‘, fcn)

disconnected

disconnected from the server

reconnected

reconnected to the server

events - hoodie.store.on(‘event‘, fcn)

change, function(event, object)

data got changed

add, function(object)

data got added

update, function(object)

data got updated

remove, function(object)

data got removed

clear, function()

data got cleared

type:change

events can be namespaced, function params same as at event

type:id:change

events can be namespaced, function params same as at event

events - hoodie.account.on(‘event‘, fcn)

signup, function(username)

user signed up

signin, function(username)

user signed in

signout, function(username)

user signed out