**Current user** (2/23/14)

**from shell** I can get current user from model

ibet.model.people.get\_user() => user object

3) **current user**:

Keep track of current user in model

**/\*** this object is reset after return from server

you need to delete the old current user before set the new current user\*/

**user object**

\_\_id : "T...03"

\_\_s: true

cid: "id\_6"

css\_map : bgd, top, left

name: "Richard"

\_proto:object

**/\*** **this is return from server**, this info use to reset current user object \*/

**user\_map :**

\_id : "id\_6"

**cid: undefined**

css\_map : bgd, top, left

name: "Richard"

\_proto:object

// this is interesting ???

stateMap.**people\_cid\_map[ user.cid ] = user**;

**current user** is save in stateMap.user in **model**

var

configMap = { anon\_id : 'a0' },

stateMap = {

anon\_user : null,

cid\_serial : 0,

is\_connected : false,

**people\_cid\_map : {},**

people\_db : TAFFY(),

**user : null**

},

**get\_user** = function () { **return stateMap.user;** };

stateMap.user is string of user name ??

personProto = {

**get\_is\_user** : function () {

/\* how I create this.cid

if this object is person, then it should has 'cid' property

\*/

return this.cid === stateMap.user.cid;

},

**get\_is\_anon** : function () {

return this.cid === stateMap.anon\_user.cid;

}

};

**people =** (function () { // people closure in model

var get\_by\_cid, get\_db, get\_user, login, logout;

get\_by\_cid = function ( cid ) {

return stateMap.people\_cid\_map[ cid ];

};

get\_db = function () { return stateMap.people\_db; };

onTapAcct = function ( event ) {

var acct\_text, user\_name, user = ibet.model.people.get\_user();

if ( user.get\_is\_anon() ) {

clearPeopleDb = function () {

**// yes stateMap.user is a person object**

var user = stateMap.user;

stateMap.people\_db = TAFFY();

stateMap.people\_cid\_map = {};

if ( user ) {

// because it inserts the user object to DB

stateMap.people\_**db.insert( user );**

// this is interesting ???

stateMap.**people\_cid\_map[ user.cid ] = user**;

}

};

/\* **user\_map object is return from server**, its info are used to reset current user object

**user object is reset after** return from server

you need to delete the old current user before set the new current user \*/

**completeLogin** = function ( user\_list ) {

var user\_map = user\_list[ 0 ];

delete stateMap.people\_cid\_map[ user\_map.cid ];

stateMap.user.cid = user\_map.\_id;

stateMap.user.id = user\_map.\_id;

stateMap.user.css\_map = user\_map.css\_map;

stateMap.people\_cid\_map[ user\_map.\_id ] = stateMap.user;

// When we add chat, we should join here

$.gevent.publish( 'ibet-login', [ stateMap.user ] );

};

-----------

Model

initModule = function () {

// initialize anonymous person

stateMap.anon\_user = makePerson({

cid : configMap.anon\_id,

id : configMap.anon\_id,

name : 'anonymous'

});

stateMap.user = stateMap.anon\_user;

};

return {

initModule : initModule,

people : people

};

from shell I can get current user from model

ibet.model.people.get\_user() => user object

**user object**

\_\_id : "T...03"

\_\_s: true

cid: "id\_6"

css\_map : bgd, top, left

name: "Richard"

\_proto:object