Social Gaming Praktikum

Summersemester 2015

Campus Conquest

Applied Programming Interface

Michael Sailer

Jonas Mayer

Paul Preißner

Jean-Paul Vieira

Benedict Drechsler

Julian Frattini

Praktikum Social Gaming

Inhaltsverzeichnis

[UserHandler 3](#_Toc424404427)

[Login 3](#_Toc424404428)

[enterFacultyIfNone 3](#_Toc424404429)

[getFaculty 4](#_Toc424404430)

[getPlayerStats 4](#_Toc424404431)

[setClasses 4](#_Toc424404432)

[getAllData 5](#_Toc424404433)

[updatePositions 6](#_Toc424404434)

[attack 6](#_Toc424404435)

[passive 7](#_Toc424404436)

[pullUpdate 7](#_Toc424404437)

[getPlayersInVicinity 7](#_Toc424404438)

[callAbility 8](#_Toc424404439)

[setActive 8](#_Toc424404440)

[setInactive 9](#_Toc424404441)

[DataHandler 10](#_Toc424404442)

[faculties 10](#_Toc424404443)

[getCapturePoints 10](#_Toc424404444)

[getBaseClasses 11](#_Toc424404445)

[getSuperClasses 11](#_Toc424404446)

[getAbilities 11](#_Toc424404447)

[GameHandler 12](#_Toc424404448)

[getPlayerCount 12](#_Toc424404449)

[getCaptureStats 12](#_Toc424404450)

[getTeamScoreboard 13](#_Toc424404451)

# UserHandler

|  |  |
| --- | --- |
| Login | |
| **Status** | verified |
| **Call** | /users/:name/:facebookID/:longitude/:latitude/login |
| **Description** | Logs the user onto the server. If he does not exist, a new account is created |
| **Parameters** | *name*: a string which is used to represent the player in the game  *facebookId*: a mixed String of fixed length gotten through the facebook login, returned by the login function and to be saved locally for future use  *longitude*: a double representing the respective value given by Google Maps  *latitude*: see longitude |
| **Returns** |  |
| **Success** | *none* |
| **Fail** | JSON FailureObject  102: longitude out of bounds |

|  |  |
| --- | --- |
| enterFacultyIfNone | |
| **Status** | verified |
| **Call** | /users/:facebookID/:faculty/enterFacultyIfNone |
| **Description** | Signs up the user to the given faculty if he hasn’t chosen one yet |
| **Parameters** | *facebookId*: a mixed String of fixed length gotten through the facebook login, returned by the login function and to be saved locally for future use  *faculty*: positive integer of an existing faculty (see getAllFaculties) |
| **Returns** |  |
| **Success** | HTTP 200 OK |
| **Fail** | JSON FailureObject  120 missing parameter  121 facebookID not found, please register first.  122 faculty not found, please provide a valid facultyId.  123 user already belongs to a faculty, can't change faculty. |

|  |  |
| --- | --- |
| getFaculty | |
| **Status** | verified |
| **Call** | /users/:facebookID/getFaculty |
| **Description** | Returns the current faculty of the player |
| **Parameters** | *facebookId*: a mixed String of fixed length gotten through the facebook login, returned by the login function and to be saved locally for future use |
| **Returns** |  |
| **Success** | facultyID : int (-1 if none) |
| **Fail** | JSON FailureObject  130 missing parameter |

|  |  |
| --- | --- |
| getPlayerStats | |
| **Status** | verified |
| **Call** | /users/:facebookID/getPlayerStats |
| **Description** | Returns all Information needed in the Statistics Fragment |
| **Parameters** | *facebookId*: a mixed String of fixed length gotten through the facebook login, returned by the login function and to be saved locally for future use |
| **Returns** |  |
| **Success** | Playtime :  Hours : integer  Minutes : integer  Seconds : integer  score : integer  kills : integer  deaths : integer  captures : integer  faculty : integer  polygons : integer |
| **Fail** | JSON FailureObject  100: facebookId not found, please register first. |

|  |  |
| --- | --- |
| setClasses | |
| **Status** | verified |
| **Call** | /users/:facebookID/:class1/:class2/setClasses |
| **Description** | Sets the two base-classes of the user with the fiven facebookID |
| **Parameters** | *facebookId*: a mixed String of fixed length gotten through the facebook login, returned by the login function and to be saved locally for future use  *class1/class2*: an integer-value between 0 and 4, representing the corresponding base-class |
| **Returns** |  |
| **Success** | *none* |
| **Fail** | JSON FailureObject  100: facebookId not found, please register first. |
|  |  |
| getAllData | |
| **Status** | verified |
| **Call** | /users/:facebookID/getAllData |
| **Description** | Returns all Data for the Player-Overview |
| **Parameters** | *facebookId*: a mixed String of fixed length gotten through the facebook login, returned by the login function and to be saved locally for future use |
| **Returns** |  |
| **Success** | Class :  class1 : integer  class2 : integer  superclass: String  cooldown1 : integer  cooldown2 : integer  Attributes :  life : integer  strength : integer  intelligence : integer  dominance : integer  sight : integer  strengthBuffed : boolean  sightBuffed : boolean  invisible : boolean  Status :  hp : integer  maxhp : integer  ExP : integer  ExPToNext : integer  level : integer  status : String (ACTIVE, INACTIVE, DEAD) |
| **Fail** | JSON FailureObject  100: facebookId not found, please register first. |

|  |  |
| --- | --- |
| updatePositions | |
| **Status** | verified |
| **Call** | /users/:facebookId/:longitude/:latitude/updatePosition |
| **Description** | Updates the players position in the map, should be called regularily.  A call of this method will put the user in an active state serverside for one minute unless the user performs a successful logout of some sort. |
| **Parameters** | *facebookId*: a mixed String of fixed length gotten through the facebook login, returned by the login function and to be saved locally for future use  *longitude*: a double representing the respective value given by Google Maps  *latitude*: see longitude |
| **Returns** |  |
| **Success** | HTTP 200 OK |
| **Fail** | JSON FailureObject  100: facebookId not found, please register first.  101: facebookId already active with another device, please logout one device.  102: longitude out of bounds.  103: latitude out of bounds.  104: Missing parameter, None of the parameter may be null. |
|  |  |
| attack | |
| **Status** | verified |
| **Call** | /users/:facebookId/attack |
| **Description** | tba |
| **Parameters** | *facebookId*: a mixed String of fixed length gotten through the facebook login, returned by the login function and to be saved locally for future use |
| **Returns** |  |
| **Success** | tba |
| **Fail** | tba |
|  |  |
| passive | |
| **Status** | implemented (not used) |
| **Call** | /users/:facebookId/passive |
| **Description** | tba |
| **Parameters** | *facebookId*: a mixed String of fixed length gotten through the facebook login, returned by the login function and to be saved locally for future use |
| **Returns** |  |
| **Success** | tba |
| **Fail** | tba |

|  |  |
| --- | --- |
| pullUpdate | |
| **Status** | verified |
| **Call** | /users/:facebookId/pullUpdate |
| **Description** | This method gives a client all information needed after a simulation step: - Attack Meshes that exist - User's health and energy |
| **Parameters** | *facebookId*: a mixed String of fixed length gotten through the facebook login, returned by the login function and to be saved locally for future use |
| **Returns** |  |
| **Success** | hp : integer  energy : integer  allMeshes :  facultyID : integer  meshes :  mesh*x* : (x = 1, 2, 3, …)  point*x* : (x = 1, 2, 3, …)  facebookID : String  x : double  y : double |
| **Fail** | none |
|  |  |
| getPlayersInVicinity | |
| **Status** | verified |
| **Call** | /users/:facebookID/ /getPlayersInVicinity |
| **Description** | Returns an array of all the players close to the user of the given facebookID |
| **Parameters** | *facebookId*: a mixed String of fixed length gotten through the facebook login, returned by the login function and to be saved locally for future use |
| **Returns** |  |
| **Success** | JSONArray:  facebookID : String  classname : String  name : String  facultyID : int  status : String (ACTIVE, INACTIVE, DEAD)  isInvisible : boolean  position :  longitude : double  latitude : double |
| **Fail** | JSON FailureObject  100: facebookId not found, please register first. |

|  |  |
| --- | --- |
| callAbility | |
| **Status** | verified |
| **Call** | /users/:facebookID/:number/callAbility |
| **Description** | Triggers an ability of the user. The Server takes care of level, cooldown, class, etc. |
| **Parameters** | *facebookId*: a mixed String of fixed length gotten through the facebook login, returned by the login function and to be saved locally for future use  *number*: as each character has one ability per class and two classes overall, this number must be 1 if the ability of class 1 shall be fired and 2 if otherwise |
| **Returns** |  |
| **Success** | cooldown1 : Integer  cooldown2 : Integer  fired : boolean |
| **Fail** | JSON FailureObject  100: user does not exist  101: user has not yet chosen a class  102: wrong number-parameter input |

|  |  |
| --- | --- |
| setActive | |
| **Status** | verified |
| **Call** | /users/:facebookID/setActive |
| **Description** | The Users state is set to ACTIVE. This means he actively participates in the game. |
| **Parameters** | *facebookId*: a mixed String of fixed length gotten through the facebook login, returned by the login function and to be saved locally for future use |
| **Returns** |  |
| **Success** | HTTP 200 OK |
| **Fail** | JSON FailureObject  141 : Player not found |

|  |  |
| --- | --- |
| setInactive | |
| **Status** | verified |
| **Call** | /users/:facebookID/setInactive |
| **Description** | The Users state is set to INACTIVE. This means he does not participate in the game, is not visible on the radar and simply irrelevant for the game until he sets his status back to ACTIVE again |
| **Parameters** | *facebookId*: a mixed String of fixed length gotten through the facebook login, returned by the login function and to be saved locally for future use |
| **Returns** |  |
| **Success** | HTTP 200 OK |
| **Fail** | JSON FailureObject  141 : Player not found |

# DataHandler

|  |  |
| --- | --- |
| faculties | |
| **Status** | verified |
| **Call** | /data/faculties |
| **Description** | returns a datastructure containing all faculties with a name and. Contains an id for further calls like become a member of faculty X. |
| **Parameters** | *none* |
| **Returns** |  |
| **Success** | faculties: Faculty[]  Faculty:  id: Integer  name: String  description: String |
| **Fail** | *none* |

|  |  |
| --- | --- |
| getCapturePoints | |
| **Status** | verified |
| **Call** | /data/getCapturePoints |
| **Description** | Returns all Information needed to construct the CapturePoints in the Client |
| **Parameters** | *none* |
| **Returns** |  |
| **Success** | nCapturePoints : int  CapturePoints:  letter : String  name : String  id : int  flag :  x : int  y : int  zone :  JSONArray:  x : int  y : int |
| **Fail** | none |

|  |  |
| --- | --- |
| getBaseClasses | |
| **Status** | verified |
| **Call** | /data/getBaseClasses |
| **Description** | Returns an array of the five basic classes, which a player can chose from |
| **Parameters** | *none* |
| **Returns** |  |
| **Success** | JSONArray:  id : int  name : String  description : String |
| **Fail** | *none* |

|  |  |
| --- | --- |
| getSuperClasses | |
| **Status** | verified |
| **Call** | /data/getSuperClasses |
| **Description** | Returns an array of the super-classes, which are a combination of two of the base-classes |
| **Parameters** | *none* |
| **Returns** |  |
| **Success** | JSONArray:  id : int  Subclasses : JSONArray:  id2 : integer  name : String |
| **Fail** | *none* |

|  |  |
| --- | --- |
| getAbilities | |
| **Status** | verified |
| **Call** | /data/getAbilities |
| **Description** | Returns an array of the abilities, which can be unlocked during the game. Each class provides one distinctive ability. |
| **Parameters** | *none* |
| **Returns** |  |
| **Success** | JSONArray:  classID : integer  name : String  description : String  cooldown : String |
| **Fail** | *none* |

# GameHandler

|  |  |
| --- | --- |
| getPlayerCount | |
| **Status** | implemented (not used) |
| **Call** | /game/getPlayerCount |
| **Description** | returns a data structure containing all registered users organized by their faculty. |
| **Parameters** | *none* |
| **Returns** |  |
| **Success** | playerCount: PlayerCount  faculties:  id: Integer  name: String  playerCount: Integer  players:  name: String  facebookID: String |
| **Fail** | *none* |

|  |  |
| --- | --- |
| getCaptureStats | |
| **Status** | verified |
| **Call** | /game/getCaptureStats |
| **Description** | Returns all Information needed to correctly display, which faculty holds or conquers a CapturePoint on the map |
| **Parameters** | *none* |
| **Returns** |  |
| **Success** | nCapturePoints : int  JSONArray:  id : int  dominatingFaculty : int  captured : boolean  progress : double |
| **Fail** | none |

|  |  |
| --- | --- |
| getTeamScoreboard | |
| **Status** | verified |
| **Call** | /game/getTeamScoreboard |
| **Description** | Returns all Information needed in the Statistics Fragment |
| **Parameters** | None |
| **Returns** |  |
| **Success** | Faculties :  id:Integer  score:integer  players\_registered: integer  players\_active: integer  players\_dead: integer  TopPlayers :  numberOfTopPlayers : integer (max 10)  scoreboard :  name : String  position : integer  facultyID : integer  score : integer |
| **Fail** | *none* |