1 SoCom API

The designed API for SoCom middleware as described in chapters **??** and **??** is listed here in more detail with parameters for each method.

1.1 Legend

Italic parameters are optional

1.2 Game API

URL Pattern: `game`
Class: `de.tud.kom.socom.components.game.GameManager`

API Method	I/O	Functionality	Parameter
addGame	I	Creates a new game	name, genre, password, masterpassword
removeGame	-	Removes a game	game, mastersecret
getGame	O	Shows information about an existing game, including all instances	game, password
addGameInstance	I	Creates an game instance	game, password, version, *description*
setGameInstanceDescription	I	Sets the description for a gameinstance	Cookies: game, password, gameversion; Stream: description
removeGameInstance	-	Removes a game instance	game, password, version
addGameContext	I	Creates a new context (e.g. game scene)	game, password, version, contextid, name
setGameContextDescription	I	Sets the description of a context	Cookies: game, password, gameversion, contextid; Stream: description
removeGameContext	-	Removes a game context	game, password, version, contextid
getGameContexts	O	Shows all contexts with their relations for a gameinstance	game, password, version
getGameContext	O	Shows a context with its relations	game, password, version, contextid
addGameContextRelation	I	Adds a relation between existing contexts	game, password, version, parent, child
removeGameContextRelation	I	Removes a relation	game, password, version, parent, child
getGameContextRelations	O	Shows the context relations of a gameinstance	game, password, version
setGameInstanceImage	I	Set the image for a gameinstance	Cookies: game, password, gameversion, extension; Stream: data
setGameContextImage	I	Set the image for a context	Cookies: game, password, gameversion, extension; Stream: data

Tabelle 1: API methods designed for game and context management (method parameters are ommited)

1.3 User API

URL Pattern: `user`
Class: `de.tud.kom.socom.components.user.UserManager`

1.4 Game Influence API

URL Pattern: `influence`
Class: `de.tud.kom.socom.components.influence.InfluenceManager`

1.5 Content API

URL Pattern: `content`
Class: `de.tud.kom.socom.components.content.ContentManager`

1.6 Achievements API

URL Pattern: `achievements`
Class: `de.tud.kom.socom.components.achievements.AchievementsManager`

1.7 Social Networks API

URL-Pattern: `social`
Klasse: `de.tud.kom.socom.components.social.SocialNetworkManager`

1.8 Profile-Informationen

If an attribute is not specified, its value is `null`.

1.9 Parameter

HTTP-POST: Cookie-parameters are expected in the cookie-header, while the stream should only contain the binary data.

API Method	I/O	Functionality	Parameter
getUser	O	Shows general information about a user	*id*
loginUser	-	Performs a login (linking a user to this session)	game, version, gamepassword, username, password
logout	-	Performs a logout	-
createUser	I	Creates a user & logs in	username, password, game, version, gamepassword, visibility
createUserWithSocialNetwork	I	Creates a user and generates a social network login url	game, version, gamepassword, visibility, network
deleteUser	-	Deletes a user account	password
becomeAdmin	-	Let the current user become administrator	mastersecret, password
getUsersGames	O	Shows a list of game instances a user is playing	-
setCurrentContext	I	Sets the currently playing context (scenes, level, mappart, . . .)	context
getVisitedContexts	O	Shows all contexts the user ever visited	-
addLog	I	Creates a game log (e.g. debug messages, errors, . . .)	type, message
addJournalEntry	I	Creates a journal entry (user ingame logs)	*visibility*, type, message
getJournalEntries	O	Shows all visible journal entries	*limit*, *offset*, *type*
getLogs	O	Shows all game logs	gamepassword, *limit*, *offset*, *type*
addTimePlayed	I	Adds time to the time played in the current context	time
resetTimePlayed	I	Resets the time played in the current context	-
setTimePlayed	I	Overwrites the time played in the current context	time
getTimePlayed	O	Shows the time played in the current context	-
createMetadata	I	Creates a key-value datapair for the user	key, value, *visibility*
updateMetadata	I	Changes a key-value datapair for the user	key, value, *visibility*
deleteMetadata	-	Removes a key-value datapair for the user	key, *deleted*
getMetadata	O	Shows existing, visible key-value datapairs for the current user	*of*
changeUsername	-	Changes the username	username, password
changeUserPassword	-	Changes the password	password, newpassword

Tabelle 2: API methods designed for user and user information management (method parameters are ommited)

API Method	I/O	Functionality	Parameter
`createInfluenceTemplate`	I	Creates a template for influences	visibility, question, type, *minchoices*[1], *maxchoices*[1], *contextid*, allowfreeanswers, *freeanswersvotable*[2], *maxlines*[3], *maxdigits*[3], *maxbytes*[4]
`createInfluence (template)`	I	Clones an influence template as new influence instance	templateid
`createInfluence`	I	Creates a new influence instance	visibility, question, type, *minchoices*[1], *maxchoices*[1], *contextid*, allowfreeanswers, *freeanswersvotable*[2], *maxlines*[3], *maxdigits*[3], *maxbytes*[4], *publish*, *message*[5], *time*[5],
`addPredefinedAnswer`	I	Adds a predefined answer	id, answer
`addPredefinedAnswerWithData`	I	Adds a predefined answer with data like images, audio, …	Cookies: id, answer, fileextension; Stream: data
`startInfluence`	-	Starts an influence & sets a timeout	id, *time*[6]
`stopInflunce`	-	Immediately stops an influence (timeout = now)	id
`fetchResult`	O	Shows the result of an influence	id
`getInfluence`	O	Shows the properties of an influence & its result	id
`changeInfluence`	-	Changes the properties of an influence	id, *question, type, minchoices, maxchoices, maxdigits, maxlines, maxbytes, visibility, allowfreeanswers, freeanswersvotable*

Tabelle 3: API methods designed for influence preparation, starting and result fetching (method parameters are ommited)

API Method	I/O	Functionality	Parameter
`createUserContent`	I	Prepares a user content	visibility, contextit, title, description, type, category, further[7]
`createGameContent`	I	Prepares a game content [8]	contextit, title, description, type, category, further[7]
`uploadContent`	I	Adds the data to a content	Cookies: contentident[9]; Stream: data
`getContentInfoForContext`	O	Shows available contents for a given context	context
`getContentInfo`	O	Shows available contents for given criteria	*contexts, since, type, title, keywords, metadata*
`downloadContent`	O	Sends the binary data of a content	contentid
`rateContent`	I	Rates a content	contentid, rating
`addComment`	I	Comments a content	contentid, message
`deleteComment`	-	Deletes a coment to a content	commentid, delete

Tabelle 4: API methods designed for content preparation, uploading, fetching, rating and discussing (method parameters are ommited)

API Method	I/O	Functionality	Parameter
addAchievement	I	Adds a new achievement	achievementname, description, categoryname, image, gamename, gamepassword, countermax, rewardpoints
getAchievement	O	Get an achievement with all its rewards	achievementname, gamename, gamepassword
removeAchievement	-	Delete achievement and all its relations	achievementname, gamepassword, gamename
addAchievementLevekl	I	Adds an achievement level	achievementname, gamename, gamepassword, countermax, rewardpoints
updateAchievementProcess	I	Update the achievement progress	achievementname, gamename, gamepassword, counter
resetAchievementProcess	-	Resets the achievement progress of the current achievement level	achievementname, gamename, gamepassword
getAchievementProcess	O	Get the achievements process	achievementname, gamename, gamepassword
addAchievementReward	I	Adds a new achievement reward	name, description, value
setAchievementReward	I	Relates a reward with an achievement	achievementname, gamename, gamepassword, rewardname. achievementlevel

Tabelle 5: API methods designed for user achievements (method parameters are ommited)

API Method	I/O	Functionality	Parameter
loginURL	O	Returns the URL for OAuth authentification with desired network	network
logout	-	Deletes required information for social network interaction	network
isLoggedIn	O	Whether a user is logged in to a social network	network
getNetworkFriends	O	Lists all friends of all connected social media applications -	
getSupportedNetworks	O	Lists all available (implemented) social media applications with name and ID	-
getProfileData	O	Lists the profile information as described in Tabelle 7	-
getProfileDataOf	O	Returns profile data of a different person (identified by parameters)	usersnid, network
getSupporter	O	lists the supporter of a post (likes)	post, network
publishOnFeed	I	Publishes the given text message on compatible social media feeds of the user	message
publishInfluenceOnFeed	I	Posts the text message and a link to the given influence to the social media feeds	message, influence
publishMediaOnFeed	I	Like publishOnFeed but with binary data (image, video, …)	message, type, extension
deletePost	-	Deletes a post preliminarily made	post, network
readPost	O	Returns a post including comments	post, network
comment	I	Adds a comment to a specific post	post, network, message
getSocomId	O	Socom user lookup using his social network identity	network, snuid
getPictureThumbnail	O	Link to a thumbnail photo	network, usersnid

Tabelle 6: API methods designed for social media profile reading and writing (method parameters are omitted)

Keyword (A-Z)	Description
about	Self description
birthday	Date of birth
education	Educational degree designation
email	Email address
favorite_athletes	Favorite athletes, comma separated (determined by connections like facebook likes)
favorite_teams	Favorite sport teams, comma separated
first_name	First name
gender	m or f for male/female
hometown	City of residence (as zip, name)
last_name	Sir name
languages	Comma-seperated list of languages
locale	Current place of residence (as zip, name)
name	Composed name
relationship_status	One of: single, relation, engaged, married, widowed
website	URL of users own website (externally)
work	Current employer name (usually a company name)

Tabelle 7: Normalized key-value pairs for users social media profile data.

Name (A-Z)	Type	Range/Values	Description
allowfreeanswers	boolean	true/false	Enables free answers for influences
answer	String	-	Answer to an influence
category	String	'hint', 'information', 'question', 'solution'	Category for contents
child	String	Valid context ID	Reference to the child context of a relation (to)
contentid	long	≥ 0	References a content
contentident	String	valid content ID	Uniquely identifies the currently uploading content
contextid/context	String	valid content-ID	References a context
contextids	String	-	Commaseperated list of context-IDs to references them
delete/deleted	int	0..3	Values bigger than 0 hide something: '0': visible (not deleted), '1': hide only, '2': abuse, '3': offense
description	String	-	Description of any entity
extension/fileextension	String	'png', 'mp3', 'ogg', ...(depends on function)	Fileextension of the currently uploading data
freeanswersvotable	boolean	true/false	If free answers should be allowed to be voted
game	String	-	Public name of the game
genre	String	-	Gamegenre
id (User)	long	≥ 0	References a user
influenceid/influence	int	≥ 0	References an influence
key	String	-	Any keyword specifying a metadata
keywords	String	-	Commaseperated list of words which must be in the description
limit	int	≥ 0	Limits the amount of output data
mastersecret	String	-	Secret (passphrase) needed for administrative functions
maxbytes	long	≥ 1	Maximal size (in bytes) of influence free answers which binary data
maxchoices	int	≥ minchoices	Maximal allowed answers to an influence
maxdigits	int	≥ 1	Maximal allowed digits for a free text answer
maxlines	int	≥ 1	Maximal allowed lines for a free text answer
message (Log/Journal Entry)	String	-	Message stored to a log
message (Social networks)	String	-	Message which should be posted in a social network
metadata	String	-	Metadata which must be contained and be equal. Format: 'key1:value1,key2:value2,...'
minchoices	int	≥ 1	Minimal allowed answers to an influence
name	String	-	Name of any entity
network	String	'Facebook', 'Google+'	Selects a social network
newpassword	String	-	Password which is the replacement for the current
of	long	Valid User ID	References a user id
offset	int	≥ 0	Offset to any output data (allows paging)
parent	String	Valid context ID	Reference to the parent context of a relation (from)
password (Game)	String	-	Game specific password
password (User)	String	-	Users private password for logins
post	String	Valid social network post id	References a social network post
publish	boolean	true/false	Whether it should be published in the social networks
question	String	-	The question of an influence
s rating	float	0..1	Rating between 0 and 1
since	long/String	≥ 0 / 'yyyy-MM-dd HH:mm:ss'	Selects only content which is newer than specified date. Either time in ms since 01.01.1970 00:00:00 or as date format 'yyyy-MM-dd HH:mm:ss'
templateid	int	valid influence template id	References an influence remplate
time (Influence)	long	≥ 0	Time in ms which an influence should be active
time (Playtime)	long	≥ 0	Amount of time to be added/setted
title	String	-	Titel of a game content
type (Content)	String	'text', 'audio', 'image', 'binary'	Type of a game content
type (Influence)	String	'text', 'audio', 'image', ...	Type of influence answers
type (Log/Journal Entry)	String	-	Any type for logs (Keyword 'all' inclose all types)
username	String	-	Public name of a user
usersnid/snuid	String	-	Social network intern id of a user
value	String	-	Value for a metadata
version/gameversion	String	-	Version of a game instance
visibility	int	0..4	'0': private, '1': friends-only, '2': public, '3': only socom users, '4': only socom users playing the same game