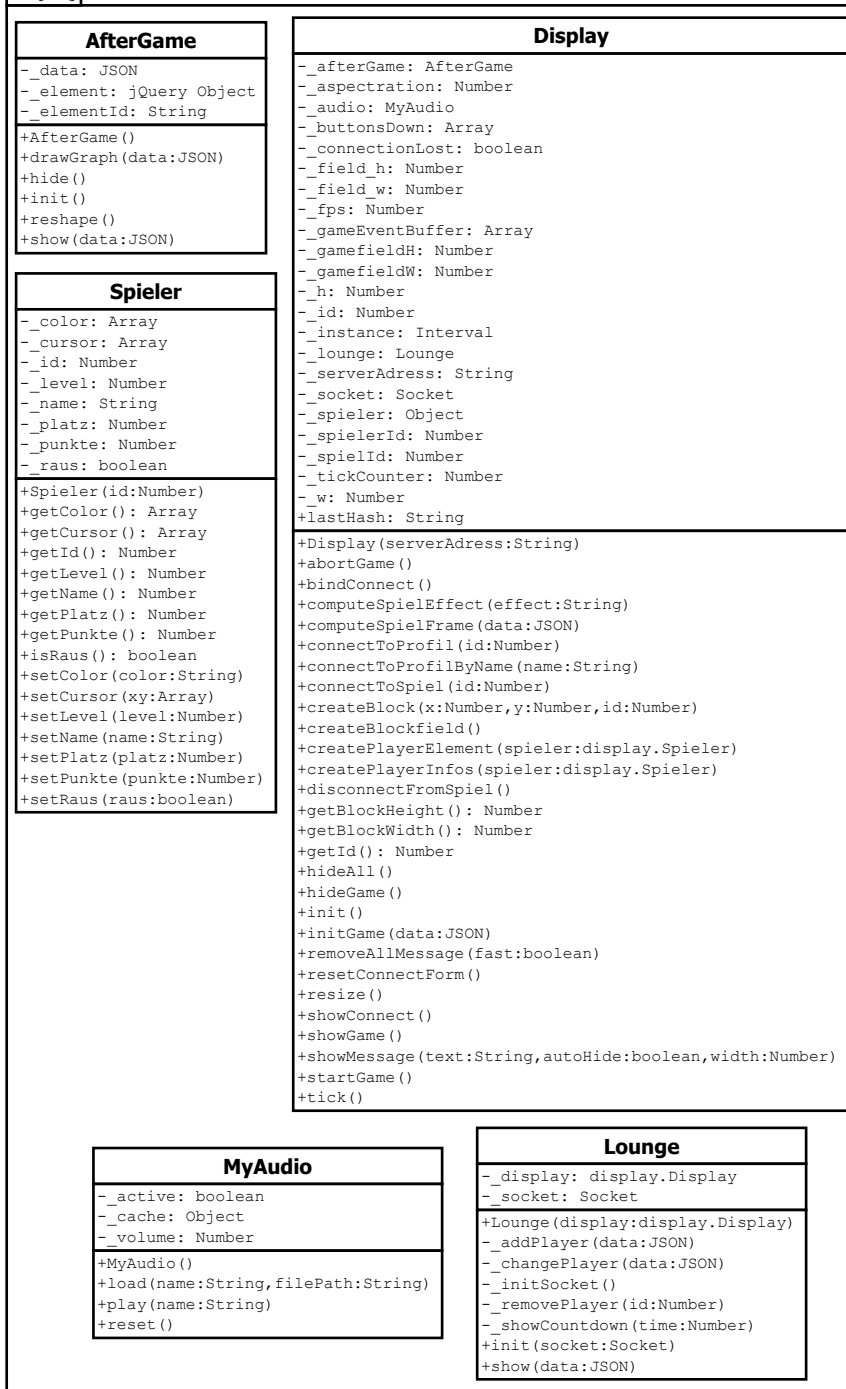
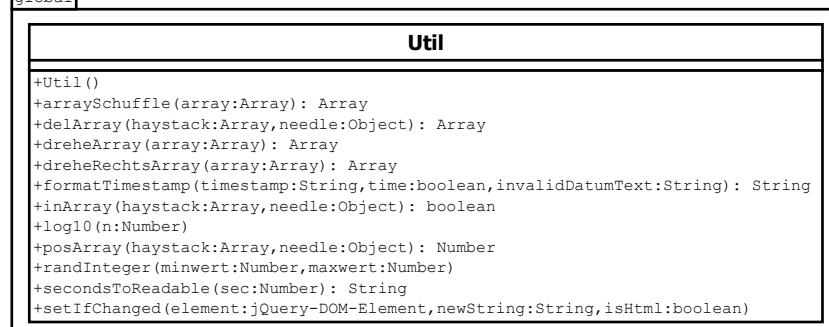


display



global



Spieler	SpielAufZeit
<pre> _active: boolean _color: Array _controller: server.Controller _cursor: Array _deleted: boolean _dieTime: boolean _elementNext: server.Games.SpielAufZeit.Blokelement _elementNow: server.Games.SpielAufZeit.Blokelement _gameinstance: server.Games.SpielAufZeit.SpielAufZeit _level: Number _lines: Number _out: boolean _platz: Number _profil: server.Controller _punkte: Number _tick: Number _tickTime: Number </pre>	<pre> _field: Array _game: server.Game _h: Number _spieler: Array _statistics: JSON _t_start: Date _w: Number *constructor: AbstractGame #SpielAufZeit (game:Game) _init_t() _t(): Date +addSpieler (profil:server.Profil,color:Array) +button (type:BUTTONTYPE, key:BUTTON, profil:server.Profil) +buttonDownDown (spieler:server.Games.SpielAufZeit.Spieler) +buttonLeftDown (spieler:server.Games.SpielAufZeit.Spieler) +buttonLeftUp (spieler:server.Games.SpielAufZeit.Spieler) +buttonRightDown (spieler:server.Games.SpielAufZeit.Spieler) +buttonRightUp (spieler:server.Games.SpielAufZeit.Spieler) +buttonRotateLeftDown (spieler:server.Games.SpielAufZeit.Spieler) +buttonRotateRightDown (spieler:server.Games.SpielAufZeit.Spieler) +calcCollision (spielerOrPosition:server.Games.SpielAufZeit.Spieler, blokelement:server.Games.SpielAufZeit.Blokelement): Number/boolean +calcPlatzierungen() +checkForLine (dest:boolean): Array +computeField() +drawOnField (xy:Array, spieler:server.Games.SpielAufZeit.Spieler, blokelement:server.Games.SpielAufZeit.Blokelement, typ:Number): Array +getActiveSpieler(): server.Games.SpielAufZeit.Spieler +getEmittedData(): JSON +getId(): Number +getI(): Number +getW(): Number +highestLine(): Number/null +init() +isXOut (x:Number): boolean +nextFrame (spieler:server.Games.SpielAufZeit.Spieler) +removeLine (yToKill:Number) +resetCursor (spieler:server.Games.SpielAufZeit.Spieler) +tick() </pre>
<pre> #Spieler (gameinstance:server.Games.SpielAufZeit.controller:server.Controller) _init() +addLevel (n:Number) +addLines (n:Number) +addPunkte (p:Number) +compute(): boolean +decrementCursorX(): Number +decrementCursorY(): Number +die() +die() +emitToController (event:String, obj:JSON) +getNextBlokelement(): server.Games.SpielAufZeit.Blokelement +getColor(): Array +getController(): server.Controller +getCursor(): Array +getCursorX(): Number +getCursorY(): Number +getEmittedData(): JSON +getId(): Number +getLevel(): Number +getLines(): Number +getNextBlokelement(): server.Games.SpielAufZeit.Blokelement </pre>	

<pre> *getProfile(): server.Profile *getPunkte(): Number *getTickTime(): Number *incrementCursorX(): Number *incrementCursorY(): Number *isActive(): boolean *nextElement() *setActive(b:boolean) *setColor(color:Array) *setCursor(ky:Array) *setCursorX(k:Number) *setCursorY(y:Number) *setLevel(n:Number) *setLines(n:Number) *setPlatz(platz:Number) </pre>	<table border="1"> <thead> <tr> <th>Block</th> <th>Blockelemente</th> </tr> </thead> <tbody> <tr> <td> <pre> ~_destroyed: boolean ~_solid: boolean ~_spieler: server.Games.SpielAufZeit.Spieler ~_vorschau: boolean </pre> </td> <td> <pre> ~_ownField: Array ~_spieler: server.Games.SpielAufZeit.Spieler ~_type: String </pre> </td> </tr> <tr> <td> <pre> *Block(spieler:server.Games.SpielAufZeit.Spieler) *getMetaData(): JSON *getSpieler(): server.Games.SpielAufZeit.Spieler *isDestroyed(): boolean *isSolid(): boolean *isVorschau(): boolean *setDestroyed(b:boolean) *setSolide(b:boolean) *setVorschau(b:boolean) </pre> </td> <td> <pre> *Blockelement(spieler:Spieler,type:String) *availableElements(): String *getType(): String *rotateLeft() *rotateRight() *setType(type:String) </pre> </td> </tr> </tbody> </table>	Block	Blockelemente	<pre> ~_destroyed: boolean ~_solid: boolean ~_spieler: server.Games.SpielAufZeit.Spieler ~_vorschau: boolean </pre>	<pre> ~_ownField: Array ~_spieler: server.Games.SpielAufZeit.Spieler ~_type: String </pre>	<pre> *Block(spieler:server.Games.SpielAufZeit.Spieler) *getMetaData(): JSON *getSpieler(): server.Games.SpielAufZeit.Spieler *isDestroyed(): boolean *isSolid(): boolean *isVorschau(): boolean *setDestroyed(b:boolean) *setSolide(b:boolean) *setVorschau(b:boolean) </pre>	<pre> *Blockelement(spieler:Spieler,type:String) *availableElements(): String *getType(): String *rotateLeft() *rotateRight() *setType(type:String) </pre>
Block	Blockelemente						
<pre> ~_destroyed: boolean ~_solid: boolean ~_spieler: server.Games.SpielAufZeit.Spieler ~_vorschau: boolean </pre>	<pre> ~_ownField: Array ~_spieler: server.Games.SpielAufZeit.Spieler ~_type: String </pre>						
<pre> *Block(spieler:server.Games.SpielAufZeit.Spieler) *getMetaData(): JSON *getSpieler(): server.Games.SpielAufZeit.Spieler *isDestroyed(): boolean *isSolid(): boolean *isVorschau(): boolean *setDestroyed(b:boolean) *setSolide(b:boolean) *setVorschau(b:boolean) </pre>	<pre> *Blockelement(spieler:Spieler,type:String) *availableElements(): String *getType(): String *rotateLeft() *rotateRight() *setType(type:String) </pre>						

```

AbstractGame
- _game: server.Game
- _spieler: Array
- _statistics: JSON
+ AbstractGame(game:server.Game)
+ addMitSpieler (profil:server.Profil,color:Array)
+ button (type:global.BUTTONTYPE, key:global.BUTTON, profil:server.Profil)
+ emitToControllers (type:String, data:JSON)
+ emitToDisplays (type:String, data:JSON)
+ getAnzahlMitSpieler () : Number
+ getMitSpieler () : Array
+ getStatistics () : JSON
+ init ()
+ removeMitprofil (profil:server.Profil)
+ tick ()

```

Game
- active: boolean
- chat: server.Chat
- colors: Array
- controller: server.Controller
- db: server.Database
- displays: Array(server.Display)
- estimator: server.Profil
- eventListeners: Array
- fireEvents: boolean
- gameCreateTime: Date
- gameEndTime: Date
- gameInstance: server.Games.AbstractGame
- gameStartTime: Date
- gameStatus: String
- interval: Interval
- lastEndTime: JSON
- maxProfil: Number
- mitProfil: Array(server.Profil)
- playerColor: Map
- profileAtStart: Array
- server: server.Server
- SpielId: Number
- SpielTyp: server.SpielTyp
- statistics: Map
- statisticsForDb: String
- tps: Number
- game (id: Number)
- getAvailableColor(): Array
- getColors(): Array
- getDataForSpielEnds(): JSON
- delete(callback:Function)
- create(callback:Function)
- getServer(server:server.Server)
- getServer(): server.Server
- getGameById(id: Number, server:server.Server,callback:Function)
- getTicksPerSecond(): Number
- getSpielTyp(): String
- save(callback:Function)
- insertScore(playerId: Number,callback:Function)
- load(callback:Function)
- hasDisplay(display:server.Display): boolean
- addDisplay(display:server.Display): boolean
- removeDisplay(display:server.Display)
- isRunning(): boolean
- addMitProfil(profile:server.Profil,force:boolean): boolean
- removeMitProfil(profile:server.Profil): boolean
- hasMitProfil(profile:server.Profil): boolean
- getMitProfil(): server.Profil
- getEstimator(profile:server.Profil)
- getEstimator(): server.Profil
- getEstimatorName(): String
- getEstimatorId(): Number
- getDuration(): Float
- getId(): Number
- getGameStatus(): String
- setGameStatus(status:String)
- startBoarding()
- getMaxProfil(): Number
- setMaxProfil(n:Numbers)
- emitToControllers(eventName:String,obj:JSON)
- emitToDisplays(eventName:String,obj:JSON)
- getSpielTypClass(): String
- createGameInstance(): server.Games.SpielAufZeit
- on(eventName: String, callback:Function)
- receiveEvent(eventName:String,obj:JSON)
- abortGame()
- endGame()
- tick()
- setSpielTyp(SpielTyp:String)
- getSpielTyp(): String
- destruct()
- addEventListener(listener:Object)
- removeEventListener(listener:Object): boolean
- stopEvents()
- startEvents(emitNow:boolean)
- broadcastEvent(eventName:String,force:boolean)
- getEditData(): JSON

```

Router

..app: express-instance
..express: express
..gameServer: server.Server
..routerCache: server.RouterCache

+Router (app:express-instance,gameServer:server.Server)
+bind404Router()
+bindApiRootRouter()
+bindRouter()
+bindControlRouter()
+bindDisplayRouter()
+bindFileRouter()
+bindHighscoreRouter()
+bindProfileRouter()
+bindRouters()
+bindStartseiteRouter()
+loadAndSendFile(req:Request,res:Response,dataPath:String,num:Number)
+setCacheParameters(req:Request)

```

Profil
-AnmeldeDatum: Date
-chat: server.Chat
-controller: server.Controller
-db: server.Database
-eventListeners: Array
-fireEvents: boolean
-gesetzteSteine: Number
-gespielteSteine: Number
-id: Number
-letzteAnmeldung: Date
-loggedInGame: server.Game
-Name: String
-online: boolean

```

create(callback:Function)
  getServer(): server.Server
  getServer(server:server.Server)
  _broadcastEvent(eventName:String,obj:JSON,
    force:boolean)
  addEventListener(listener:Object)
  addEventListener(listener:(Number))
  addEventListener(listener:(Number))
  getChat(): server.Chat
  getController(): server.Controller
  getMetaData(): JSON
  getID(): Number
  getLoggedGame(): server.Game
  getName(): String
  hasChat(): boolean
  hasController(): boolean
  isLoggedGame(): boolean
  isOnline(): boolean
  load(callback:Function)
  onMutation(data:Object)
  removeEventListener(listener:Object): boolean
  save(callback:Function)
  setChat(chat:server.Chat)
  setController(controller:server.Controller)
  setLoggedGame(game:server.Game)
  setName(name:String)
  setOnline(online:boolean)
  startEvents(emitter:NodeJS.EventEmitter)
  stopEvents()
  unsetChat()
  unsetController(bubble:boolean)

```

```

Database
- _conn: Connection
- _mysql: mysql
Database(host:String,user:String,password:String)
- _connectDB(host:String,user:String,password:String): Connection
+ connectDatabase(database:String,create_database:boolean)
+ escape(value:String)
+ query(sql:String,params:Array,caliback:Funktion)

```

```

CommandLine
- main: server.Main
+CommandLine(main:server.Main)
- interpretCommand(command:String)
+init()

```

Api
- _bindings: Map
- _gameServer: server.Server
+Api(gameServer:server.Server)
+getBindings(): Map

```

RouterCache {
-   fileCache: Map
-   minifyData: boolean
-   minifyFunctions: boolean
-   replacers: Array
-   zip: boolean
+
+ RouterCache() {
+   -doReplaces(data:JSON,ext:String)
+   -getCacheKey(pfad:String)
+   -getMinifyFunctionByExt(ext:String): Function | null
+   -getMinifyFunctions()
+   +addReplacer(replacer:Function)
+   +getData(pfad:String): JSON
+   +getMime(pfad:String)
+   +getMime(pfad:String)
+   +inCache(pfad:String): boolean
+   +loadFile(dataPath:String,callback:Function)
+   +setParams(zip:boolean,minify:boolean)
}

```

```

Display
- db: server.Database
- id: Number
- _profil: server.Profil
- _server: Server
- _socket: Socket
- _spiel: server.Game

+Display(id:Number,socket:Socket)
+destruct()
+emitToDisplay(eventName:String,obj:JSObject):Display
+getId():Number
+receiveEvent(eventName:String,obj:JSObject):server.Profil
+setProfil(profil:server.Profil)
+setSocket(socket:Socket)
+setSpiel(spiel:server.Game)

+tick()
+unsetProfil()
+unsetSocket()
+unsetSpiel()

```

```

Server
- controllers: Array
- db: server.Database
- displays: Array
- games: Array
- httpserver: Object
- idPool: Number
- io: socket.io-instance
- profiles: Array
- socket: Socket
- spieltypes: Array

server(httpserver:Object)
addProfile(profile:Profile)
addSpiel(spiel:server.Game)
createGame(data:JSON,fn:Function)
createProfile(data:JSON,fn:Function,controller:Function)
deleteSpiel(spiel:server.Game)
emitToAllCoControllers(eventName:String,obj:JSON)
getAllGames(): Array
getDB(): server.Database
getProfileById(id:Number): server.Profile | null
getSpielById(id:Number): server.Game | null
getSpielTypById(id:Number): server.Spieltyp | null
receiveEvent(eventName:String,obj:JSON)
removeSpiel(spiel:server.Game)
restart()
start()
stop()

```

```

SpieleTyp
bezeichnung: String
class: String
fireEvents: boolean
id: Number

spieleTyp(id: Number, bezeichnung: String, cals: String)
addEventListener(listener: Object)
roadcastEvent(force: boolean)
etBezeichnung(): String
etClass(): String
etEmitData(): Object
etId(): Number
emoveEventListener()
etBezeichnung(bezeichnung: String)
startEvents()
topEvents()

```

Chat
-_active: boolean
-_game: server.Game
-_messages: Array
+Chat(game:server.Game)
+isActive(): boolean
+setActive(active:boolean)
+write(profile:server.Profil,message:String)

```

Controller
- _db: server.Database
- _id: Number
- _profil: server.Profil
- _server: server
- _socket: Socket

@ControlId: number, socket: Socket
getServer(): server.Server
$setServer(server: server.Server)
$destroy()
emitToController(eventName: String, obj: JSON)
getId(): Number
hasProfil(): server.Profil
hasProfil(): boolean
setProfil(profil: server.Profil)
setSocket(socket: Socket)
tick()
userProfil(bubble: boolean)

```

Auswahl
<div><div>-_socket: Socket</div><div>-_controller: controller.Controller</div><div>-_isTouch: boolean</div><div>-_chat: controller.Chat</div></div>
<div><div>+Auswahl()</div><div>+init(controller:controller.Controller)</div><div>+createProfil(data:JSON,callback:Funktion)</div><div>+createGame(data:JOSN,callback:Funktion)</div><div>+showLevelauswahl()</div><div>+showProfilauswahl()</div><div>+showLounge(data:JSON)</div><div>-_bindSocketEvents()</div><div>+updateProfillist(callback:Funktion)</div><div>+updateGamelist(callback:Funktion)</div><div>+addGameToSelection(row:JSON,statisch:boolean=false)</div><div>+changeGameInSelection(data:JSON)</div><div>+deleteGameInSelection(data:JSON)</div><div>+addProfilToSelection(row:JSON,statisch:boolean=true)</div><div>+changeProfilInSelection(data:JSON)</div><div>+deleteProfilInSelection(data:JSON)</div><div>+submitCreateGame()</div><div>+submitCreateProfil()</div><div>+abmelden()</div><div>+selectLounge(SpielID:Number)</div><div>+disselectLounge(SpielID:Number)</div><div>+startGameFromLounge(SpielID:Number)</div><div>+showCreateGame()</div><div>+cancelCreateGame()</div><div>+showCreateProfil()</div><div>+cancelCreateProfil()</div><div>+selectProfil(ProfilID:Number)</div><div>-_initListeners()</div><div>-_htmlGetRoundButton(text:String,clas:String)</div><div>-_htmlMakeLoungeList(MaxProfil:Number,ProfilIn:Array)</div><div>+flash()</div><div>+show()</div><div>+hide()</div><div>+backToMain()</div></div>

Controller
<div><div>-_active: boolean</div><div>-_auswahl: controller.Auswahl</div><div>-_buttonsDown: Array</div><div>-_catchKeyboard: boolean</div><div>-_color: Array</div><div>-_connectionLost: boolean</div><div>-_elesMap: Map</div><div>-_game_selected: Number</div><div>-_id: Number</div><div>-_isTouch: boolean</div><div>-_keyMap: Map</div><div>-_profil: String</div><div>-_profil_id: Number</div><div>-_serverAdress: String</div><div>-_socket: Socket</div><div>-_touch_deactivated: boolean</div></div>
<div><div>+Controller(serverAdress)</div><div>-_bindSocketEvents()</div><div>-_initListeners()</div><div>-_initSocket(callback:Funktion)</div><div>-_reconnected()</div><div>-_resize()</div><div>+buttonDown(button:BUTTON)</div><div>+buttonUp(button:BUTTON)</div><div>+connectControllerToServer(callback:Funktion)</div><div>+connectionLost()</div><div>+deselectSpiel()</div><div>+disselectProfil(callback:Funktion)</div><div>+getGameSelected(): Number</div><div>+getId(): Number</div><div>+getProfil(): JSON</div><div>+getProfilId(): Number</div><div>+getSocket(): Socket</div><div>+hide()</div><div>+init(callback:Funktion,auswahl:contoller.Auswahl)</div><div>+isActive(): boolean</div><div>+leaveGame(callback:Funktion,spielId:Number)</div><div>+selectGame(callback:Funktion,spielId:Number)</div><div>+selectProfil(callback:Funktion,profilId:Number)</div><div>+setActive(b:boolean)</div><div>+setColor(color:Array)</div><div>+setProfil(profil:String)</div><div>+setProfilId(id:Number)</div><div>+show(time:Number)</div><div>+startGame(callback:Funktion,spiel_id:Number)</div></div>

Chat
<div><div>-_rootID: Number</div><div>-_name: String</div><div>-_socket: Socket</div></div>
<div><div>+Chat(socket:Socket)</div><div>-_bindSocketEvents()</div><div>+add(von:String,text:String)</div><div>+clear()</div><div>+delegate()</div><div>+getRootHtml()</div><div>+getRootId()</div><div>+setName(name:String)</div><div>+getRootId(rootId)</div><div>+write(text:String)</div></div>