

GameCreator

Generated by Doxygen 1.8.20



<b>1 Hierarchical Index</b>	<b>1</b>
1.1 Class Hierarchy	1
<b>2 Class Index</b>	<b>3</b>
2.1 Class List	3
<b>3 Class Documentation</b>	<b>5</b>
3.1 Area Class Reference	5
3.1.1 Detailed Description	5
3.1.2 Member Function Documentation	5
3.1.2.1 CreatePageData()	5
3.2 inja::AstNode Class Reference	6
3.2.1 Detailed Description	6
3.3 inja::BlockNode Class Reference	7
3.4 Category Class Reference	7
3.4.1 Detailed Description	7
3.4.2 Member Function Documentation	8
3.4.2.1 CreatePageData()	8
3.5 inja::Environment Class Reference	8
3.5.1 Detailed Description	9
3.5.2 Member Function Documentation	9
3.5.2.1 include_template()	9
3.6 inja::ExpressionListNode Class Reference	10
3.7 inja::ExpressionNode Class Reference	10
3.8 inja::FileError Struct Reference	11
3.9 inja::ForArrayStatementNode Class Reference	11
3.10 inja::ForObjectStatementNode Class Reference	12
3.11 inja::ForStatementNode Class Reference	12
3.12 inja::FunctionStorage::FunctionData Struct Reference	13
3.13 inja::FunctionNode Class Reference	13
3.14 inja::FunctionStorage Class Reference	14
3.14.1 Detailed Description	15
3.15 inja::IfStatementNode Class Reference	15
3.16 inja::IncludeStatementNode Class Reference	15
3.17 inja::InjaError Struct Reference	16
3.18 inja::JsonError Struct Reference	17
3.19 inja::JsonNode Class Reference	17
3.20 inja::Lexer Class Reference	18
3.20.1 Detailed Description	18
3.21 inja::LiteralNode Class Reference	18
3.22 inja::NodeVisitor Class Reference	19
3.23 Page Class Reference	20
3.23.1 Detailed Description	20

3.23.2 Member Function Documentation	20
3.23.2.1 CreatePageData()	20
3.23.2.2 GenerateChildNodes()	21
3.23.2.3 GenerateParentNodes()	21
3.23.3 Member Data Documentation	21
3.23.3.1 child_nodes_	21
3.23.3.2 parent_nodes_	21
3.24 inja::Parser Class Reference	22
3.24.1 Detailed Description	22
3.25 inja::ParserError Struct Reference	22
3.26 inja::Renderer Class Reference	23
3.26.1 Detailed Description	23
3.27 inja::RenderError Struct Reference	23
3.28 ServerFrame Class Reference	24
3.28.1 Detailed Description	25
3.28.2 Constructor & Destructor Documentation	25
3.28.2.1 ServerFrame()	25
3.28.3 Member Function Documentation	25
3.28.3.1 AddElem()	25
3.28.3.2 Backups()	25
3.28.3.3 CheckRunning()	26
3.28.3.4 CreateRequest()	26
3.28.3.5 DelElem()	26
3.28.3.6 DelUser()	27
3.28.3.7 DoLogin()	27
3.28.3.8 DoLogout()	27
3.28.3.9 DoRegistration()	27
3.28.3.10 GetLog()	28
3.28.3.11 GrantAccessTo()	28
3.28.3.12 IsRunning()	28
3.28.3.13 LoginPage()	29
3.28.3.14 ServeFile()	29
3.28.3.15 Start()	29
3.28.3.16 StartGame()	30
3.28.3.17 WriteObject()	30
3.29 inja::SetStatementNode Class Reference	30
3.30 inja::SourceLocation Struct Reference	31
3.31 inja::StatementNode Class Reference	31
3.32 inja::StatisticsVisitor Class Reference	32
3.32.1 Detailed Description	32
3.33 inja::Template Struct Reference	32
3.33.1 Detailed Description	33

3.34 inja::TextNode Class Reference . . . . .	33
3.35 inja::Token Struct Reference . . . . .	33
3.35.1 Detailed Description . . . . .	34
3.36 User Class Reference . . . . .	34
3.36.1 Detailed Description . . . . .	35
3.36.2 Constructor & Destructor Documentation . . . . .	35
3.36.2.1 User() [1/2] . . . . .	35
3.36.2.2 User() [2/2] . . . . .	35
3.36.3 Member Function Documentation . . . . .	36
3.36.3.1 AddFile() . . . . .	36
3.36.3.2 AddLocation() . . . . .	36
3.36.3.3 AddNewObject() . . . . .	37
3.36.3.4 AddRequest() . . . . .	37
3.36.3.5 CheckAccessToLocations() . . . . .	38
3.36.3.6 CheckGameRunning() . . . . .	38
3.36.3.7 CreateBackup() . . . . .	38
3.36.3.8 CreateNewWorld() . . . . .	39
3.36.3.9 DeleteBackup() . . . . .	39
3.36.3.10 DeleteFile() . . . . .	39
3.36.3.11 DeleteObject() . . . . .	40
3.36.3.12 DeleteWorld() . . . . .	40
3.36.3.13 GetBackups() . . . . .	40
3.36.3.14 GetCategory() . . . . .	41
3.36.3.15 GetObjects() . . . . .	41
3.36.3.16 GetOverview() . . . . .	41
3.36.3.17 GetPortOfWorld() . . . . .	41
3.36.3.18 GetWorld() . . . . .	42
3.36.3.19 RemoveRequest() . . . . .	42
3.36.3.20 RestoreBackup() . . . . .	42
3.36.3.21 SafeUser() . . . . .	43
3.36.3.22 WriteObject() . . . . .	43
3.37 UserManager Class Reference . . . . .	43
3.37.1 Detailed Description . . . . .	44
3.37.2 Constructor & Destructor Documentation . . . . .	44
3.37.2.1 UserManager() . . . . .	44
3.37.3 Member Function Documentation . . . . .	44
3.37.3.1 DeleteUser() . . . . .	44
3.37.3.2 DoLogin() . . . . .	45
3.37.3.3 DoLogout() . . . . .	45
3.37.3.4 DoRegistration() . . . . .	45
3.37.3.5 GenerateCookie() . . . . .	47
3.37.3.6 GetAllWorlds() . . . . .	47

---

3.37.3.7	GetNextPort()	47
3.37.3.8	GetSharedWorlds()	48
3.37.3.9	GetUser()	48
3.37.3.10	GetUserFromCookie()	48
3.37.3.11	GrantAccessTo()	49
3.38	World Class Reference	49
3.38.1	Detailed Description	49
3.38.2	Constructor & Destructor Documentation	49
3.38.2.1	World()	49
3.38.2.2	~World()	50
3.38.3	Member Function Documentation	50
3.38.3.1	GetPage()	50
3.39	Worlds Class Reference	50
3.39.1	Detailed Description	51
3.39.2	Constructor & Destructor Documentation	51
3.39.2.1	Worlds()	51
3.39.2.2	~Worlds()	51
3.39.3	Member Function Documentation	51
3.39.3.1	GetPage()	51

<b>Index</b>	<b>53</b>
--------------	-----------

# Chapter 1

## Hierarchical Index

### 1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

inja::AstNode . . . . .	6
inja::BlockNode . . . . .	7
inja::ExpressionListNode . . . . .	10
inja::ExpressionNode . . . . .	10
inja::FunctionNode . . . . .	13
inja::JsonNode . . . . .	17
inja::LiteralNode . . . . .	18
inja::StatementNode . . . . .	31
inja::ForStatementNode . . . . .	12
inja::ForArrayStatementNode . . . . .	11
inja::ForObjectStatementNode . . . . .	12
inja::IfStatementNode . . . . .	15
inja::IncludeStatementNode . . . . .	15
inja::SetStatementNode . . . . .	30
inja::TextNode . . . . .	33
inja::Environment . . . . .	8
inja::FunctionStorage::FunctionData . . . . .	13
inja::FunctionStorage . . . . .	14
inja::Lexer . . . . .	18
inja::NodeVisitor . . . . .	19
inja::Renderer . . . . .	23
inja::StatisticsVisitor . . . . .	32
Page . . . . .	20
Area . . . . .	5
Category . . . . .	7
inja::Parser . . . . .	22
runtime_error	
inja::InjaError . . . . .	16
inja::FileError . . . . .	11
inja::JsonError . . . . .	17
inja::ParserError . . . . .	22
inja::RenderError . . . . .	23
ServerFrame . . . . .	24
inja::SourceLocation . . . . .	31

inja::Template . . . . .	32
inja::Token . . . . .	33
User . . . . .	34
UserManager . . . . .	43
World . . . . .	49
Worlds . . . . .	50



## Chapter 2

# Class Index

### 2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">Area</a>	5
<a href="#">inja::AstNode</a>	
Base node class for the abstract syntax tree (AST)	6
<a href="#">inja::BlockNode</a>	7
<a href="#">Category</a>	7
<a href="#">inja::Environment</a>	
Class for changing the configuration	8
<a href="#">inja::ExpressionListNode</a>	10
<a href="#">inja::ExpressionNode</a>	10
<a href="#">inja::FileError</a>	11
<a href="#">inja::ForArrayStatementNode</a>	11
<a href="#">inja::ForObjectStatementNode</a>	12
<a href="#">inja::ForStatementNode</a>	12
<a href="#">inja::FunctionStorage::FunctionData</a>	13
<a href="#">inja::FunctionNode</a>	13
<a href="#">inja::FunctionStorage</a>	
Class for builtin functions and user-defined callbacks	14
<a href="#">inja::IfStatementNode</a>	15
<a href="#">inja::IncludeStatementNode</a>	15
<a href="#">inja::InjaError</a>	16
<a href="#">inja::JsonError</a>	17
<a href="#">inja::JsonNode</a>	17
<a href="#">inja::Lexer</a>	
Class for lexing an <a href="#">inja Template</a>	18
<a href="#">inja::LiteralNode</a>	18
<a href="#">inja::NodeVisitor</a>	19
<a href="#">Page</a>	20
<a href="#">inja::Parser</a>	
Class for parsing an <a href="#">inja Template</a>	22
<a href="#">inja::ParserError</a>	22
<a href="#">inja::Renderer</a>	
Class for rendering a <a href="#">Template</a> with data	23
<a href="#">inja::RenderError</a>	23
<a href="#">ServerFrame</a>	24
<a href="#">inja::SetStatementNode</a>	30

<a href="#">inja::SourceLocation</a>	31
<a href="#">inja::StatementNode</a>	31
<a href="#">inja::StatisticsVisitor</a>	
A class for counting statistics on a <a href="#">Template</a>	32
<a href="#">inja::Template</a>	
The main inja <a href="#">Template</a>	32
<a href="#">inja::TextNode</a>	33
<a href="#">inja::Token</a>	
Helper-class for the inja <a href="#">Lexer</a>	33
<a href="#">User</a>	34
<a href="#">UserManager</a>	43
<a href="#">World</a>	49
<a href="#">Worlds</a>	50

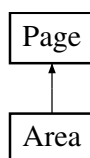
## Chapter 3

# Class Documentation

### 3.1 Area Class Reference

```
#include <area.h>
```

Inheritance diagram for Area:



#### Public Member Functions

- **Area** (std::string base\_path\_, std::string path, nlohmann::json objects)
- nlohmann::json [CreatePageData](#) (std::string path)

#### Additional Inherited Members

##### 3.1.1 Detailed Description

Author

flux

##### 3.1.2 Member Function Documentation

###### 3.1.2.1 CreatePageData()

```
nlohmann::json Area::CreatePageData (
    std::string path ) [virtual]
```

Calls base-class function for area and CreateObjectPageData for object.

**Parameters**

<code>in</code>	<code>path</code>	to area or object.
-----------------	-------------------	--------------------

**Returns**

json with information.

Reimplemented from [Page](#).

The documentation for this class was generated from the following files:

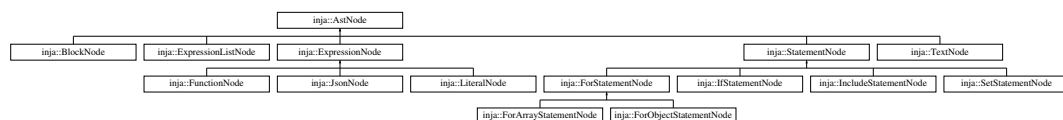
- `/home/fux/Documents/programming/Der-Zug/src/gui/src/page/area.h`
- `/home/fux/Documents/programming/Der-Zug/src/gui/src/page/area.cc`

## 3.2 inja::AstNode Class Reference

Base node class for the abstract syntax tree (AST).

```
#include <inja.hpp>
```

Inheritance diagram for inja::AstNode:

**Public Member Functions**

- virtual void **accept** ([NodeVisitor](#) &v) const =0
- **AstNode** (size\_t pos)

**Public Attributes**

- size\_t **pos**

### 3.2.1 Detailed Description

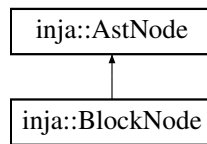
Base node class for the abstract syntax tree (AST).

The documentation for this class was generated from the following file:

- `/home/fux/Documents/programming/Der-Zug/src/gui/src/users/inja.hpp`

## 3.3 inja::BlockNode Class Reference

Inheritance diagram for inja::BlockNode:



### Public Member Functions

- void **accept** ([NodeVisitor](#) &v) const

### Public Attributes

- std::vector< std::shared\_ptr< [AstNode](#) > > **nodes**

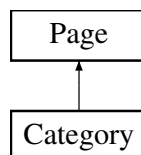
The documentation for this class was generated from the following file:

- /home/fux/Documents/programming/Der-Zug/src/gui/src/users/inja.hpp

## 3.4 Category Class Reference

```
#include <category.h>
```

Inheritance diagram for Category:



### Public Member Functions

- **Category** (std::string base\_path, std::string path)
- nlohmann::json [CreatePageData](#) (std::string path)

### Additional Inherited Members

#### 3.4.1 Detailed Description

Author

fux Class for all categories. As category is a page refering to other categories not objects.

### 3.4.2 Member Function Documentation

#### 3.4.2.1 CreatePageData()

```
nlohmann::json Category::CreatePageData (
    std::string path ) [virtual]
```

Calls base-class function.

##### Parameters

in	<i>path</i>	to area or object.
----	-------------	--------------------

##### Returns

json with information.

Reimplemented from [Page](#).

The documentation for this class was generated from the following files:

- /home/fux/Documents/programming/Der-Zug/src/gui/src/page/category.h
- /home/fux/Documents/programming/Der-Zug/src/gui/src/page/category.cc

## 3.5 inja::Environment Class Reference

Class for changing the configuration.

```
#include <inja.hpp>
```

### Public Member Functions

- **Environment** (const std::string &global\_path)
- **Environment** (const std::string &input\_path, const std::string &output\_path)
- void [set\\_statement](#) (const std::string &open, const std::string &close)
 

*Sets the opener and closer for template statements.*
- void [set\\_line\\_statement](#) (const std::string &open)
 

*Sets the opener for template line statements.*
- void [set\\_expression](#) (const std::string &open, const std::string &close)
 

*Sets the opener and closer for template expressions.*
- void [set\\_comment](#) (const std::string &open, const std::string &close)
 

*Sets the opener and closer for template comments.*
- void [set\\_trim\\_blocks](#) (bool trim\_blocks)
 

*Sets whether to remove the first newline after a block.*
- void [set\\_lstrip\\_blocks](#) (bool lstrip\_blocks)

- Sets whether to strip the spaces and tabs from the start of a line to a block.*
  - void `set_search_included_templates_in_files` (bool search\_in\_files)
- Sets the element notation syntax.*
  - void `set_throw_at_missing_includes` (bool will\_throw)
- Sets whether a missing include will throw an error.*
  - `Template` `parse` (nonstd::string\_view input)
  - `Template` `parse_template` (const std::string &filename)
  - `Template` `parse_file` (const std::string &filename)
  - std::string `render` (nonstd::string\_view input, const json &data)
  - std::string `render` (const `Template` &tmpl, const json &data)
  - std::string `render_file` (const std::string &filename, const json &data)
  - std::string `render_file_with_json_file` (const std::string &filename, const std::string &filename\_data)
  - void `write` (const std::string &filename, const json &data, const std::string &filename\_out)
  - void `write` (const `Template` &temp, const json &data, const std::string &filename\_out)
  - void `write_with_json_file` (const std::string &filename, const std::string &filename\_data, const std::string &filename\_out)
  - void `write_with_json_file` (const `Template` &temp, const std::string &filename\_data, const std::string &filename\_out)
  - std::ostream & `render_to` (std::ostream &os, const `Template` &tmpl, const json &data)
  - std::string `load_file` (const std::string &filename)
  - json `load_json` (const std::string &filename)
  - void `add_callback` (const std::string &name, const CallbackFunction &callback)
- Adds a variadic callback.*
  - void `add_void_callback` (const std::string &name, const VoidCallbackFunction &callback)
- Adds a variadic void callback.*
  - void `add_callback` (const std::string &name, int num\_args, const CallbackFunction &callback)
- Adds a callback with given number or arguments.*
  - void `add_void_callback` (const std::string &name, int num\_args, const VoidCallbackFunction &callback)
- Adds a void callback with given number or arguments.*
  - void `include_template` (const std::string &name, const `Template` &tmpl)

### 3.5.1 Detailed Description

Class for changing the configuration.

### 3.5.2 Member Function Documentation

#### 3.5.2.1 include\_template()

```
void inja::Environment::include_template (
    const std::string & name,
    const Template & tmpl ) [inline]
```

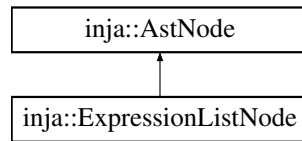
Includes a template with a given name into the environment. Then, a template can be rendered in another template using the include "<name>" syntax.

The documentation for this class was generated from the following file:

- /home/fux/Documents/programming/Der-Zug/src/gui/src/users/inja.hpp

## 3.6 inja::ExpressionListNode Class Reference

Inheritance diagram for inja::ExpressionListNode:



### Public Member Functions

- **ExpressionListNode** (size\_t pos)
- void **accept** (NodeVisitor &v) const

### Public Attributes

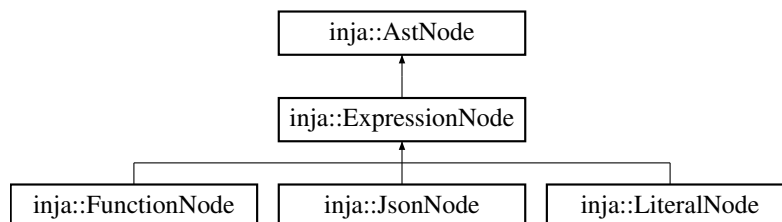
- std::vector< std::shared\_ptr< ExpressionNode > > **rpn\_output**

The documentation for this class was generated from the following file:

- /home/fux/Documents/programming/Der-Zug/src/gui/src/users/inja.hpp

## 3.7 inja::ExpressionNode Class Reference

Inheritance diagram for inja::ExpressionNode:



### Public Member Functions

- **ExpressionNode** (size\_t pos)
- void **accept** (NodeVisitor &v) const

### Additional Inherited Members

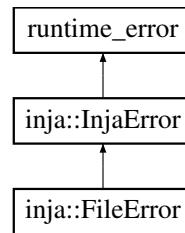
The documentation for this class was generated from the following file:

- /home/fux/Documents/programming/Der-Zug/src/gui/src/users/inja.hpp



## 3.8 inja::FileError Struct Reference

Inheritance diagram for inja::FileError:



### Public Member Functions

- **FileError** (const std::string &message)
- **FileError** (const std::string &message, [SourceLocation](#) location)

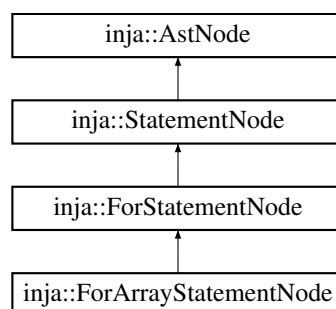
### Additional Inherited Members

The documentation for this struct was generated from the following file:

- /home/fux/Documents/programming/Der-Zug/src/gui/src/users/inja.hpp

## 3.9 inja::ForArrayStatementNode Class Reference

Inheritance diagram for inja::ForArrayStatementNode:



### Public Member Functions

- **ForArrayStatementNode** (const std::string &value, [BlockNode](#) \*const parent, size\_t pos)
- void **accept** ([NodeVisitor](#) &v) const

## Public Attributes

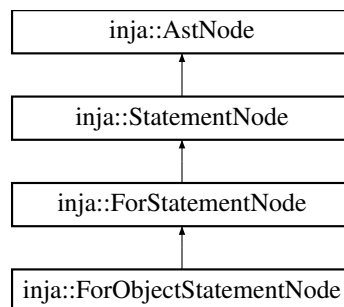
- `const std::string value`

The documentation for this class was generated from the following file:

- `/home/fux/Documents/programming/Der-Zug/src/gui/src/users/inja.hpp`

## 3.10 inja::ForObjectStatementNode Class Reference

Inheritance diagram for `inja::ForObjectStatementNode`:



## Public Member Functions

- **ForObjectStatementNode** (`const std::string &key`, `const std::string &value`, `BlockNode *const parent`, `size_t pos`)
- `void accept (NodeVisitor &v) const`

## Public Attributes

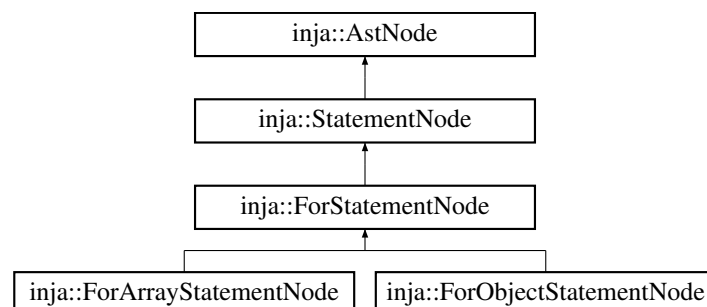
- `const std::string key`
- `const std::string value`

The documentation for this class was generated from the following file:

- `/home/fux/Documents/programming/Der-Zug/src/gui/src/users/inja.hpp`

## 3.11 inja::ForStatementNode Class Reference

Inheritance diagram for `inja::ForStatementNode`:



## Public Member Functions

- **ForStatementNode** ([BlockNode](#) \*const parent, size\_t pos)
- virtual void **accept** ([NodeVisitor](#) &v) const =0

## Public Attributes

- [ExpressionListNode](#) **condition**
- [BlockNode](#) **body**
- [BlockNode](#) \*const **parent**

The documentation for this class was generated from the following file:

- /home/fux/Documents/programming/Der-Zug/src/gui/src/users/inja.hpp

## 3.12 inja::FunctionStorage::FunctionData Struct Reference

### Public Member Functions

- **FunctionData** (const Operation &op, const CallbackFunction &cb=CallbackFunction{})

### Public Attributes

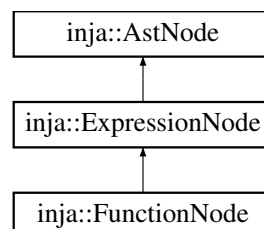
- const Operation **operation**
- const CallbackFunction **callback**

The documentation for this struct was generated from the following file:

- /home/fux/Documents/programming/Der-Zug/src/gui/src/users/inja.hpp

## 3.13 inja::FunctionNode Class Reference

Inheritance diagram for inja::FunctionNode:



### Public Types

- enum **Associativity** { **Left**, **Right** }

## Public Member Functions

- **FunctionNode** (nonstd::string\_view name, size\_t pos)
- **FunctionNode** (Op operation, size\_t pos)
- void **accept** ([NodeVisitor](#) &v) const

## Public Attributes

- unsigned int **precedence**
- Associativity **associativity**
- Op **operation**
- std::string **name**
- int **number\_args**
- CallbackFunction **callback**

The documentation for this class was generated from the following file:

- /home/fux/Documents/programming/Der-Zug/src/gui/src/users/inja.hpp

## 3.14 inja::FunctionStorage Class Reference

Class for builtin functions and user-defined callbacks.

```
#include <inja.hpp>
```

## Classes

- struct [FunctionData](#)

## Public Types

- enum **Operation** {  
**Not, And, Or, In,**  
**Equal, NotEqual, Greater, GreaterEqual,**  
**Less, LessEqual, Add, Subtract,**  
**Multiplication, Division, Power, Modulo,**  
**AtId, At, Default, DivisibleBy,**  
**Even, Exists, ExistsInObject, First,**  
**Float, Int, IsArray, IsBoolean,**  
**IsFloat, IsInteger, IsNumber, IsObject,**  
**IsString, Last, Length, Lower,**  
**Max, Min, Odd, Range,**  
**Round, Sort, Upper, Callback,**  
**ParenLeft, ParenRight, None }**

## Public Member Functions

- void **add\_builtin** (nonstd::string\_view name, int num\_args, Operation op)
- void **add\_callback** (nonstd::string\_view name, int num\_args, const CallbackFunction &callback)
- [FunctionData](#) **find\_function** (nonstd::string\_view name, int num\_args) const

### 3.14.1 Detailed Description

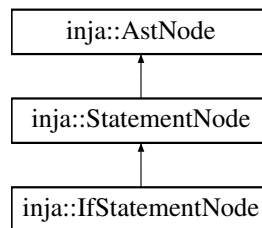
Class for builtin functions and user-defined callbacks.

The documentation for this class was generated from the following file:

- /home/fux/Documents/programming/Der-Zug/src/gui/src/users/inja.hpp

## 3.15 inja::IfStatementNode Class Reference

Inheritance diagram for inja::IfStatementNode:



### Public Member Functions

- **IfStatementNode** ([BlockNode](#) \*const parent, size\_t pos)
- **IfStatementNode** (bool is\_nested, [BlockNode](#) \*const parent, size\_t pos)
- void **accept** ([NodeVisitor](#) &v) const

### Public Attributes

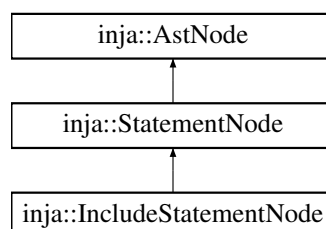
- [ExpressionListNode](#) **condition**
- [BlockNode](#) **true\_statement**
- [BlockNode](#) **false\_statement**
- [BlockNode](#) \*const **parent**
- const bool **is\_nested**
- bool **has\_false\_statement** {false}

The documentation for this class was generated from the following file:

- /home/fux/Documents/programming/Der-Zug/src/gui/src/users/inja.hpp

## 3.16 inja::IncludeStatementNode Class Reference

Inheritance diagram for inja::IncludeStatementNode:



## Public Member Functions

- **IncludeStatementNode** (const std::string &file, size\_t pos)
- void **accept** ([NodeVisitor](#) &v) const

## Public Attributes

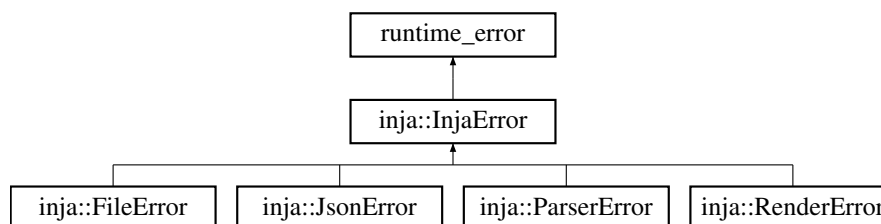
- const std::string **file**

The documentation for this class was generated from the following file:

- /home/fux/Documents/programming/Der-Zug/src/gui/src/users/inja.hpp

## 3.17 inja::InjaError Struct Reference

Inheritance diagram for inja::InjaError:



## Public Member Functions

- **InjaError** (const std::string &type, const std::string &message)
- **InjaError** (const std::string &type, const std::string &message, [SourceLocation](#) location)

## Public Attributes

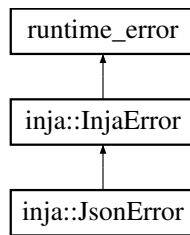
- const std::string **type**
- const std::string **message**
- const [SourceLocation](#) **location**

The documentation for this struct was generated from the following file:

- /home/fux/Documents/programming/Der-Zug/src/gui/src/users/inja.hpp

## 3.18 inja::JsonError Struct Reference

Inheritance diagram for inja::JsonError:



### Public Member Functions

- **JsonError** (const std::string &message, [SourceLocation](#) location)

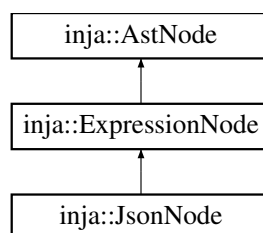
### Additional Inherited Members

The documentation for this struct was generated from the following file:

- /home/fux/Documents/programming/Der-Zug/src/gui/src/users/inja.hpp

## 3.19 inja::JsonNode Class Reference

Inheritance diagram for inja::JsonNode:



### Public Member Functions

- **JsonNode** (nonstd::string\_view ptr\_name, size\_t pos)
- void **accept** ([NodeVisitor](#) &v) const

### Static Public Member Functions

- static std::string **convert\_dot\_to\_json\_ptr** (nonstd::string\_view ptr\_name)

## Public Attributes

- const std::string **name**
- const json::json\_pointer **ptr**

The documentation for this class was generated from the following file:

- /home/fux/Documents/programming/Der-Zug/src/gui/src/users/inja.hpp

## 3.20 inja::Lexer Class Reference

Class for lexing an inja [Template](#).

```
#include <inja.hpp>
```

### Public Member Functions

- **Lexer** (const LexerConfig &config)
- [SourceLocation](#) **current\_position** () const
- void **start** (nonstd::string\_view input)
- [Token](#) **scan** ()
- const LexerConfig & **get\_config** () const

### 3.20.1 Detailed Description

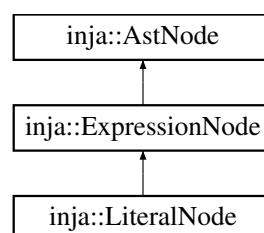
Class for lexing an inja [Template](#).

The documentation for this class was generated from the following file:

- /home/fux/Documents/programming/Der-Zug/src/gui/src/users/inja.hpp

## 3.21 inja::LiteralNode Class Reference

Inheritance diagram for inja::LiteralNode:





## Public Member Functions

- **LiteralNode** (const nlohmann::json &value, size\_t pos)
- void **accept** ([NodeVisitor](#) &v) const

## Public Attributes

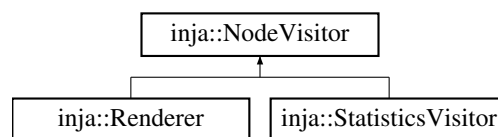
- const nlohmann::json **value**

The documentation for this class was generated from the following file:

- /home/fux/Documents/programming/Der-Zug/src/gui/src/users/inja.hpp

## 3.22 inja::NodeVisitor Class Reference

Inheritance diagram for inja::NodeVisitor:



## Public Member Functions

- virtual void **visit** (const [BlockNode](#) &node)=0
- virtual void **visit** (const [TextNode](#) &node)=0
- virtual void **visit** (const [ExpressionNode](#) &node)=0
- virtual void **visit** (const [LiteralNode](#) &node)=0
- virtual void **visit** (const [JsonNode](#) &node)=0
- virtual void **visit** (const [FunctionNode](#) &node)=0
- virtual void **visit** (const [ExpressionListNode](#) &node)=0
- virtual void **visit** (const [StatementNode](#) &node)=0
- virtual void **visit** (const [ForStatementNode](#) &node)=0
- virtual void **visit** (const [ForArrayStatementNode](#) &node)=0
- virtual void **visit** (const [ForObjectStatementNode](#) &node)=0
- virtual void **visit** (const [IfStatementNode](#) &node)=0
- virtual void **visit** (const [IncludeStatementNode](#) &node)=0
- virtual void **visit** (const [SetStatementNode](#) &node)=0

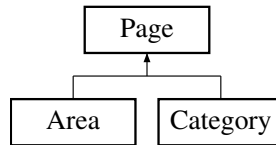
The documentation for this class was generated from the following file:

- /home/fux/Documents/programming/Der-Zug/src/gui/src/users/inja.hpp

## 3.23 Page Class Reference

```
#include <page.h>
```

Inheritance diagram for Page:



### Public Member Functions

- **Page** (std::string base\_path, std::string path)
- virtual nlohmann::json [CreatePageData](#) (std::string path)

### Protected Member Functions

- void [GenerateParentNodes](#) ()
- virtual void [GenerateChildNodes](#) ()

### Protected Attributes

- std::string **base\_path\_**
- std::string **path\_**
- std::string **name\_**
- std::map< std::string, std::string > [child\\_nodes\\_](#)
- std::map< std::string, std::string > [parent\\_nodes\\_](#)

### 3.23.1 Detailed Description

Author

flux Base class for pages. Class is abstract and never instantiated. Also storing all parents and child-nodes.

### 3.23.2 Member Function Documentation

#### 3.23.2.1 CreatePageData()

```
nlohmann::json Page::CreatePageData (
    std::string path ) [virtual]
```

Creates data for page: json-data & path to matching template. Used by category and area, however not for objects.

**Parameters**

<i>in</i>	<i>path</i>	path to target page.
-----------	-------------	----------------------

**Returns**

json with information.

Reimplemented in [Category](#), and [Area](#).

**3.23.2.2 GenerateChildNodes()**

```
virtual void Page::GenerateChildNodes ( ) [inline], [protected], [virtual]
```

Updates all child-nodes. (Virtual add different for category/ area.)

**3.23.2.3 GenerateParentNodes()**

```
void Page::GenerateParentNodes ( ) [protected]
```

Generates all parents. (Identical for category and area.)

**3.23.3 Member Data Documentation****3.23.3.1 child\_nodes\_**

```
std::map<std::string, std::string> Page::child_nodes_ [protected]
```

Child nodes. key: url-path, value: child-name.

**3.23.3.2 parent\_nodes\_**

```
std::map<std::string, std::string> Page::parent_nodes_ [protected]
```

Parent nodes. key: url-path, value: parent-name.

The documentation for this class was generated from the following files:

- /home/fux/Documents/programming/Der-Zug/src/gui/src/page/page.h
- /home/fux/Documents/programming/Der-Zug/src/gui/src/page/page.cc

## 3.24 inja::Parser Class Reference

Class for parsing an inja [Template](#).

```
#include <inja.hpp>
```

### Public Member Functions

- **Parser** (const ParserConfig &parser\_config, const LexerConfig &lexer\_config, TemplateStorage &template\_storage, const [FunctionStorage](#) &function\_storage)
- [Template](#) **parse** (nonstd::string\_view input, nonstd::string\_view path)
- [Template](#) **parse** (nonstd::string\_view input)
- void **parse\_into\_template** ([Template](#) &tmpl, nonstd::string\_view filename)
- std::string **load\_file** (nonstd::string\_view filename)

### 3.24.1 Detailed Description

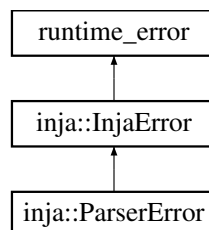
Class for parsing an inja [Template](#).

The documentation for this class was generated from the following file:

- /home/fux/Documents/programming/Der-Zug/src/gui/src/users/inja.hpp

## 3.25 inja::ParserError Struct Reference

Inheritance diagram for inja::ParserError:



### Public Member Functions

- **ParserError** (const std::string &message, [SourceLocation](#) location)

### Additional Inherited Members

The documentation for this struct was generated from the following file:

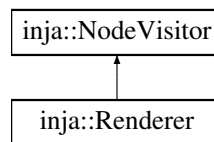
- /home/fux/Documents/programming/Der-Zug/src/gui/src/users/inja.hpp

## 3.26 inja::Renderer Class Reference

Class for rendering a [Template](#) with data.

```
#include <inja.hpp>
```

Inheritance diagram for inja::Renderer:



### Public Member Functions

- **Renderer** (const RenderConfig &config, const TemplateStorage &template\_storage, const [FunctionStorage](#) &function\_storage)
- void **render\_to** (std::ostream &os, const [Template](#) &tmpl, const json &data, json \*loop\_data=nullptr)

#### 3.26.1 Detailed Description

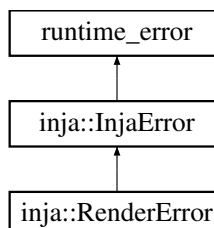
Class for rendering a [Template](#) with data.

The documentation for this class was generated from the following file:

- /home/fux/Documents/programming/Der-Zug/src/gui/src/users/inja.hpp

## 3.27 inja::RenderError Struct Reference

Inheritance diagram for inja::RenderError:



### Public Member Functions

- **RenderError** (const std::string &message, [SourceLocation](#) location)

## Additional Inherited Members

The documentation for this struct was generated from the following file:

- /home/fux/Documents/programming/Der-Zug/src/gui/src/users/inja.hpp

## 3.28 ServerFrame Class Reference

```
#include <server_frame.h>
```

### Public Member Functions

- [ServerFrame](#) ()
- [UserManager](#) & [user\\_manager](#) ()
- void [Start](#) (int port)
- void [LoginPage](#) (const httpplib::Request &req, httpplib::Response &resp) const
- void [DoLogin](#) (const httpplib::Request &req, httpplib::Response &resp)
- void [DoRegistration](#) (const httpplib::Request &req, httpplib::Response &resp)
- void [DoLogout](#) (const httpplib::Request &req, httpplib::Response &resp)
- void [DelUser](#) (const httpplib::Request &req, httpplib::Response &resp)
  - Deletes a user. [User](#) is deleted on own request.*
- void [ServeFile](#) (const httpplib::Request &req, httpplib::Response &resp, bool backup=false) const
- void [AddElem](#) (const httpplib::Request &req, httpplib::Response &resp)
  - Add new element (world, subcategory, object).*
- void [DelElem](#) (const httpplib::Request &req, httpplib::Response &resp)
  - Delete element (world, subcategory, object).*
- void [WriteObject](#) (const httpplib::Request &req, httpplib::Response &resp)
  - Write (modified) object.*
- void [Backups](#) (const httpplib::Request &req, httpplib::Response &resp, std::string action)
  - Creates a backup of the selected world.*
- void [GrantAccessTo](#) (const httpplib::Request &req, httpplib::Response &resp)
  - Grants given user access to world of other user.*
- void [CreateRequest](#) (const httpplib::Request &req, httpplib::Response &resp)
  - Grants given user access to world of other user.*
- void [CheckRunning](#) (const httpplib::Request &req, httpplib::Response &resp)
  - Checks whether game is still running.*
- void [GetLog](#) (const httpplib::Request &req, httpplib::Response &resp)
  - Checks whether game is still running.*
- void [StartGame](#) (const httpplib::Request &req, httpplib::Response &resp)
  - Checks whether game is still running.*
- bool [IsRunning](#) ()
  - Gives feedback on whether server is still running*
- void [Stop](#) ()
  - Makes server stop running.*
- [~ServerFrame](#) ()
  - Destructor, which stops server.*

### 3.28.1 Detailed Description

Author

flux

### 3.28.2 Constructor & Destructor Documentation

#### 3.28.2.1 ServerFrame()

```
ServerFrame::ServerFrame ( )
```

Constructor. Creating user\_manager with path to user-data and possible categories.

### 3.28.3 Member Function Documentation

#### 3.28.3.1 AddElem()

```
void ServerFrame::AddElem (
    const httpplib::Request & req,
    httpplib::Response & resp )
```

Add new element (world, subcategory, object).

Parameters

in	<i>req</i>	(reference to request)
in, out	<i>resp</i>	(reference to response)

#### 3.28.3.2 Backups()

```
void ServerFrame::Backups (
    const httpplib::Request & req,
    httpplib::Response & resp,
    std::string action )
```

Creates a backup of the selected world.

Parameters

in	<i>req</i>	(reference to request)
in, out	<i>resp</i>	(reference to response)

Generated by Doxygen

### 3.28.3.3 CheckRunning()

```
void ServerFrame::CheckRunning (
    const httpplib::Request & req,
    httpplib::Response & resp )
```

Checks whether game is still running.

#### Parameters

in	<i>req</i>	(reference to request)
in, out	<i>resp</i>	(reference to response)

### 3.28.3.4 CreateRequest()

```
void ServerFrame::CreateRequest (
    const httpplib::Request & req,
    httpplib::Response & resp )
```

Grants given user access to world of other user.

#### Parameters

in	<i>req</i>	(reference to request)
in, out	<i>resp</i>	(reference to response)

### 3.28.3.5 DelElem()

```
void ServerFrame::DelElem (
    const httpplib::Request & req,
    httpplib::Response & resp )
```

Delete element (world, subcategory, object).

#### Parameters

in	<i>req</i>	(reference to request)
in, out	<i>resp</i>	(reference to response)



### 3.28.3.6 DelUser()

```
void ServerFrame::DelUser (
    const httpplib::Request & req,
    httpplib::Response & resp )
```

Deletes a user. [User](#) is deleted on own request.

#### Parameters

in	<i>req</i>	(reference to request)
in, out	<i>resp</i>	(reference to response)

### 3.28.3.7 DoLogin()

```
void ServerFrame::DoLogin (
    const httpplib::Request & req,
    httpplib::Response & resp )
```

function to handle login

#### Parameters

in	<i>req</i>	(reference to request)
in, out	<i>resp</i>	(server response)
in	<i>user_manager</i>	(user_manager to check whether user exists)

### 3.28.3.8 DoLogout()

```
void ServerFrame::DoLogout (
    const httpplib::Request & req,
    httpplib::Response & resp )
```

logout a user

#### Parameters

in	<i>req</i>	(reference to request)
in, out	<i>resp</i>	(server response)

### 3.28.3.9 DoRegistration()

```
void ServerFrame::DoRegistration (
```

```
const httpplib::Request & req,
httpplib::Response & resp )
```

function to handle registration

#### Parameters

in	<i>req</i>	(reference to request)
in, out	<i>resp</i>	(server response)

### 3.28.3.10 GetLog()

```
void ServerFrame::GetLog (
    const httpplib::Request & req,
    httpplib::Response & resp )
```

Checks whether game is still running.

#### Parameters

in	<i>req</i>	(reference to request)
in, out	<i>resp</i>	(reference to response)

### 3.28.3.11 GrantAccessTo()

```
void ServerFrame::GrantAccessTo (
    const httpplib::Request & req,
    httpplib::Response & resp )
```

Grants given user access to world of other user.

#### Parameters

in	<i>req</i>	(reference to request)
in, out	<i>resp</i>	(reference to response)

### 3.28.3.12 IsRunning()

```
bool ServerFrame::IsRunning ( )
```

Gives feedback on whether server is still running

**Returns**

boolean

**3.28.3.13 LoginPage()**

```
void ServerFrame::LoginPage (
    const httpplib::Request & req,
    httpplib::Response & resp ) const
```

Function to send login page, or overview page, depending on status: logged in/ not logged in.

**Parameters**

in	<i>req</i>	(reference to request)
in, out	<i>resp</i>	(server response)

**3.28.3.14 ServeFile()**

```
void ServerFrame::ServeFile (
    const httpplib::Request & req,
    httpplib::Response & resp,
    bool backup = false ) const
```

Function to send all pages (except login and start page). Redirect to login depending on status: logged in/ not logged in.

**Parameters**

in	<i>req</i>	(reference to request)
in, out	<i>resp</i>	(server response)

**3.28.3.15 Start()**

```
void ServerFrame::Start (
    int port )
```

Starts server. Server is started on given port. Also handlers are initialized.

**Parameters**

in	<i>port</i>	(Port to start server on)
----	-------------	---------------------------

### 3.28.3.16 StartGame()

```
void ServerFrame::StartGame (
    const httpplib::Request & req,
    httpplib::Response & resp )
```

Checks whether game is still running.

#### Parameters

in	<i>req</i>	(reference to request)
in, out	<i>resp</i>	(reference to response)

### 3.28.3.17 WriteObject()

```
void ServerFrame::WriteObject (
    const httpplib::Request & req,
    httpplib::Response & resp )
```

Write (modified) object.

#### Parameters

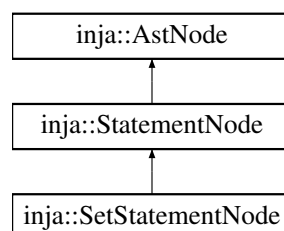
in	<i>req</i>	(reference to request)
in, out	<i>resp</i>	(reference to response)

The documentation for this class was generated from the following files:

- /home/fux/Documents/programming/Der-Zug/src/gui/src/server/server\_frame.h
- /home/fux/Documents/programming/Der-Zug/src/gui/src/server/server\_frame.cc

## 3.29 inja::SetStatementNode Class Reference

Inheritance diagram for inja::SetStatementNode:



## Public Member Functions

- **SetStatementNode** (const std::string &key, size\_t pos)
- void **accept** ([NodeVisitor](#) &v) const

## Public Attributes

- const std::string **key**
- [ExpressionListNode](#) **expression**

The documentation for this class was generated from the following file:

- /home/fux/Documents/programming/Der-Zug/src/gui/src/users/inja.hpp

## 3.30 inja::SourceLocation Struct Reference

### Public Attributes

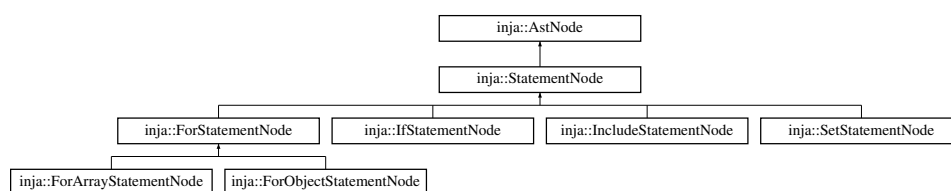
- size\_t **line**
- size\_t **column**

The documentation for this struct was generated from the following file:

- /home/fux/Documents/programming/Der-Zug/src/gui/src/users/inja.hpp

## 3.31 inja::StatementNode Class Reference

Inheritance diagram for inja::StatementNode:



## Public Member Functions

- **StatementNode** (size\_t pos)
- virtual void **accept** ([NodeVisitor](#) &v) const =0

## Additional Inherited Members

The documentation for this class was generated from the following file:

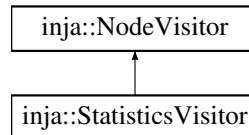
- /home/fux/Documents/programming/Der-Zug/src/gui/src/users/inja.hpp

## 3.32 inja::StatisticsVisitor Class Reference

A class for counting statistics on a [Template](#).

```
#include <inja.hpp>
```

Inheritance diagram for inja::StatisticsVisitor:



### Public Attributes

- unsigned int **variable\_counter**

### Additional Inherited Members

#### 3.32.1 Detailed Description

A class for counting statistics on a [Template](#).

The documentation for this class was generated from the following file:

- /home/fux/Documents/programming/Der-Zug/src/gui/src/users/inja.hpp

## 3.33 inja::Template Struct Reference

The main inja [Template](#).

```
#include <inja.hpp>
```

### Public Member Functions

- **Template** (const std::string &content)
- int [count\\_variables](#) ()  
*Return number of variables (total number, not distinct ones) in the template.*

### Public Attributes

- [BlockNode](#) **root**
- std::string **content**

### 3.33.1 Detailed Description

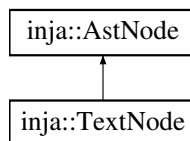
The main inja [Template](#).

The documentation for this struct was generated from the following file:

- `/home/fux/Documents/programming/Der-Zug/src/gui/src/users/inja.hpp`

## 3.34 inja::TextNode Class Reference

Inheritance diagram for inja::TextNode:



### Public Member Functions

- **TextNode** (size\_t pos, size\_t length)
- void **accept** ([NodeVisitor](#) &v) const

### Public Attributes

- const size\_t **length**

The documentation for this class was generated from the following file:

- `/home/fux/Documents/programming/Der-Zug/src/gui/src/users/inja.hpp`

## 3.35 inja::Token Struct Reference

Helper-class for the inja [Lexer](#).

```
#include <inja.hpp>
```

### Public Types

- enum **Kind** {  
**Text**, **ExpressionOpen**, **ExpressionClose**, **LineStatementOpen**,  
**LineStatementClose**, **StatementOpen**, **StatementClose**, **CommentOpen**,  
**CommentClose**, **Id**, **Number**, **String**,  
**Plus**, **Minus**, **Times**, **Slash**,  
**Percent**, **Power**, **Comma**, **Dot**,  
**Colon**, **LeftParen**, **RightParen**, **LeftBracket**,  
**RightBracket**, **LeftBrace**, **RightBrace**, **Equal**,  
**NotEqual**, **GreaterThan**, **GreaterEqual**, **LessThan**,  
**LessEqual**, **Unknown**, **Eof** }

## Public Member Functions

- constexpr **Token** (Kind kind, nonstd::string\_view text)
- std::string **describe** () const

## Public Attributes

- Kind **kind** {Kind::Unknown}
- nonstd::string\_view **text**

### 3.35.1 Detailed Description

Helper-class for the inja [Lexer](#).

The documentation for this struct was generated from the following file:

- /home/fux/Documents/programming/Der-Zug/src/gui/src/users/inja.hpp

## 3.36 User Class Reference

```
#include <user.h>
```

## Public Member Functions

- [User](#) (std::string username, std::string pw, std::string path, std::vector< std::string > categories)
- [User](#) (std::string username, std::string pw, std::string path, std::vector< std::string > locations, std::vector< std::string > categories)
- const std::string **username** () const
- std::string **password** () const
- std::map< std::string, int > & **worlds** ()
- const std::vector< std::string > & **locations** () const
- void **set\_password** (std::string password)
- void [AddLocation](#) (std::string user, std::string world)
- std::string [GetOverview](#) (nlohmann::json shared\_worlds, nlohmann::json all\_worlds)
- std::string [GetWorld](#) (std::string path, std::string user, std::string world, int port)  
*Get Overview of categories in one world.*
- std::string [GetCategory](#) (std::string path, std::string user, std::string world, std::string category)
- std::string [GetBackups](#) (std::string user, std::string world)
- std::string [GetObjects](#) (std::string path, std::string user, std::string world, std::string category, std::string sub)
- std::string **GetObject** (std::string path, std::string user, std::string world, std::string category, std::string sub, std::string obj)
- int [CreateNewWorld](#) (std::string name, int port)  
*Creates a new world for this user.*
- int [AddFile](#) (std::string path, std::string name)  
*Adds new file to category Adds new file and tries to create json matching category.*
- int [AddNewObject](#) (std::string path, std::string id, bool force=false)  
*Adds a new empty object.*
- int [WriteObject](#) (std::string request)



- int [DeleteWorld](#) (std::string world)
- int [DeleteFile](#) (std::string path, std::string subcategory)
- int [DeleteObject](#) (std::string path, std::string object)
- void [SafeUser](#) () const
- bool [CheckAccessToLocations](#) (std::string path)
- int [CreateBackup](#) (std::string user, std::string world)
- int [RestoreBackup](#) (std::string user, std::string backup)
- int [DeleteBackup](#) (std::string user, std::string backup)
- bool [CheckGameRunning](#) (std::string path)
- int [GetPortOfWorld](#) (std::string path)
- bool [AddRequest](#) (std::string user, std::string world)
- void [RemoveRequest](#) (std::string user, std::string world)

### 3.36.1 Detailed Description

Author

flux

### 3.36.2 Constructor & Destructor Documentation

#### 3.36.2.1 User() [1/2]

```
User::User (
    std::string name,
    std::string pw,
    std::string path,
    std::vector< std::string > cats )
```

Constructor creating a (new) user.

Parameters

in	<i>username.</i>	
in	<i>pw</i>	(password).
in	<i>path</i>	(path to folders).
in	<i>categories</i>	(all currently available categories)

Author

flux

#### 3.36.2.2 User() [2/2]

```
User::User (
    std::string username,
```

```

std::string pw,
std::string path,
std::vector< std::string > locations,
std::vector< std::string > categories )

```

Constructor creating a (new) user.

#### Parameters

in	<i>username.</i>	
in	<i>pw</i>	(password).
in	<i>path</i>	(path to folders).
in	<i>locations</i>	(locations available for this user).
in	<i>categories</i>	(all currently available categories)

### 3.36.3 Member Function Documentation

#### 3.36.3.1 AddFile()

```

int User::AddFile (
    std::string path,
    std::string name )

```

Adds new file to category Adds new file and tries to create json matching category.

#### Parameters

<i>path</i>	
<i>name</i>	

#### Returns

error code.

#### 3.36.3.2 AddLocation()

```

void User::AddLocation (
    std::string user,
    std::string world )

```

Add a new location. Adds a new location (a new world) to list of locations by adding path: "[new\_↵username]/files/[new\_world]" to locations.

## Parameters

in	<i>user</i>	(user who granted access)
in	<i>world</i>	(new world which is added)

**3.36.3.3 AddNewObject()**

```
int User::AddNewObject (
    std::string path,
    std::string id,
    bool force = false )
```

Adds a new empty object.

## Parameters

<i>path</i>	
<i>id</i>	
<i>force</i>	

## Returns

ErrorCode.

**3.36.3.4 AddRequest()**

```
bool User::AddRequest (
    std::string user,
    std::string world )
```

Adds a new access-request.

## Parameters

in	<i>user</i>	(user who asks for access)
in	<i>world</i>	(world to gran access for)

## Returns

SUCCESS.

**3.36.3.5 CheckAccessToLocations()**

```
bool User::CheckAccessToLocations (
    std::string path )
```

Check whether user has access to given path.

**Parameters**

in	<i>path</i> .	
----	---------------	--

**3.36.3.6 CheckGameRunning()**

```
bool User::CheckGameRunning (
    std::string path )
```

Tries to run game. Extracts user and world from path and tries to run game with all existing players.

**Parameters**

in	<i>path</i>	(path to extract information from)
in	<i>number</i>	(of players to check)

**Returns**

boolean indicating success.

**3.36.3.7 CreateBackup()**

```
int User::CreateBackup (
    std::string user,
    std::string world )
```

Create a backup of given world.

**Parameters**

in	<i>user</i>	
in	<i>world</i>	

**Returns**

ErrorCode.

### 3.36.3.8 CreateNewWorld()

```
int User::CreateNewWorld (
    std::string name,
    int port )
```

Creates a new world for this user.

#### Parameters

<i>name</i>	(Name of the world)
<i>port</i>	

#### Returns

Success code.

### 3.36.3.9 DeleteBackup()

```
int User::DeleteBackup (
    std::string user,
    std::string backup )
```

Delete a backup of given world.

#### Parameters

in	<i>user</i>	
in	<i>backup</i>	

#### Returns

ErrorCode.

### 3.36.3.10 DeleteFile()

```
int User::DeleteFile (
    std::string path,
    std::string subcategory )
```

delete a given file (subcategory)

#### Parameters

in	<i>path</i>	(path to category)
in	<i>subcategory</i>	return ErrorCode.

### 3.36.3.11 DeleteObject()

```
int User::DeleteObject (
    std::string path,
    std::string object )
```

delete a given object

#### Parameters

in	<i>path</i>	(path to json)
in	<i>object</i>	(object to delete from json) return ErrorCode.

### 3.36.3.12 DeleteWorld()

```
int User::DeleteWorld (
    std::string world )
```

delete a given file (subcategory)

#### Parameters

in	<i>world</i>	return ErrorCode.
----	--------------	-------------------

### 3.36.3.13 GetBackups()

```
std::string User::GetBackups (
    std::string user,
    std::string world )
```

Get Overview of a backups. (f.e. world1/rooms = trainstation, hospital...)

#### Parameters

in	<i>user</i>	
in	<i>world</i>	

#### Returns

overview page of all backups

**3.36.3.14 GetCategory()**

```
std::string User::GetCategory (
    std::string path,
    std::string user,
    std::string world,
    std::string category )
```

Get Overview of a category. (f.e. world1/rooms = trainstation, hospital...)

**3.36.3.15 GetObjects()**

```
std::string User::GetObjects (
    std::string path,
    std::string user,
    std::string world,
    std::string category,
    std::string sub )
```

Get Overview of a SubCategory. (f.e. world1/rooms/trainstation = platform a, great\_hall ...)

**3.36.3.16 GetOverview()**

```
std::string User::GetOverview (
    nlohmann::json shared_worlds,
    nlohmann::json all_worlds )
```

Gets all worlds and parses them into the overview page.

**Parameters**

in	<i>shared_worlds</i>	(json with all shared_worlds and ports)
in	<i>all_worlds</i>	(json with all worlds and ports)

**Returns**

overview-page.

**3.36.3.17 GetPortOfWorld()**

```
int User::GetPortOfWorld (
    std::string path )
```

Get a attribute of a certain world

**Parameters**

in	<i>path</i>	(path to world)
in	<i>attr</i>	

### 3.36.3.18 GetWorld()

```
std::string User::GetWorld (
    std::string path,
    std::string user,
    std::string world,
    int port )
```

Get Overview of categories in one world.

#### Parameters

<i>path</i>	(path to current world)
<i>world</i>	(name of accessed world)
<i>port</i>	(port of world, as it might not be the current users port)

#### Returns

HTML page or empty string.

### 3.36.3.19 RemoveRequest()

```
void User::RemoveRequest (
    std::string user,
    std::string world )
```

Removes an access-request.

#### Parameters

in	<i>user</i>	(user who asks for access)
in	<i>world</i>	(world to gran access for)

#### Returns

success.

### 3.36.3.20 RestoreBackup()

```
int User::RestoreBackup (
    std::string user,
    std::string backup )
```

Restores a backup of given world.



**Parameters**

in	<i>user</i>	
in	<i>backup</i>	(given backup to restore)

**Returns**

ErrorCode.

**3.36.3.21 SafeUser()**

```
void User::SafeUser ( ) const
```

Write users jsons to disc.

**3.36.3.22 WriteObject()**

```
int User::WriteObject (
    std::string request )
```

Write json to disc. Used to create a new object or overwrite an existsing object.

**Parameters**

in	<i>request</i>	(request to create new, or change existsing)
in	<i>force</i>	(If true, then writing is set even if game is not running.)

**Returns**

ErrorCode.

The documentation for this class was generated from the following files:

- /home/fux/Documents/programming/Der-Zug/src/gui/src/users/user.h
- /home/fux/Documents/programming/Der-Zug/src/gui/src/users/user.cc

**3.37 UserManager Class Reference**

```
#include <user_manager.h>
```

## Public Member Functions

- [UserManager](#) (std::string path, std::vector< std::string > categories)
- [User \\* GetUser](#) (std::string username) const
- void [DeleteUser](#) (std::string username)
- bool [DoLogout](#) (const char \*ptr)
- std::string [DoLogin](#) (std::string username, std::string password)
- std::string [DoRegistration](#) (std::string username, std::string pw1, std::string pw2)
- std::string [GenerateCookie](#) (std::string username)
- std::string [GetUserFromCookie](#) (const char \*ptr) const
- int [GrantAccessTo](#) (std::string user1, std::string user2, std::string world)
- int [GetNextPort](#) ()
- nlohmann::json [GetAllWorlds](#) (std::string username) const
- nlohmann::json [GetSharedWorlds](#) (std::string username) const
- int [GetPortOfWorld](#) (std::string user, std::string world) const
- std::string [GetPage](#) (std::string path) const

### 3.37.1 Detailed Description

#### Author

flux class storing all users. TODO (flux): replace this class with a database.

### 3.37.2 Constructor & Destructor Documentation

#### 3.37.2.1 UserManager()

```
UserManager::UserManager (
    std::string main_path,
    std::vector< std::string > cats )
```

constructor loading all users stored on disc.

#### Author

flux

### 3.37.3 Member Function Documentation

#### 3.37.3.1 DeleteUser()

```
void UserManager::DeleteUser (
    std::string username )
```

delete user. deletes object, erases from map and delete users jsons file.

## Parameters

in	<i>username</i>	
----	-----------------	--

**3.37.3.2 DoLogin()**

```
std::string UserManager::DoLogin (
    std::string username,
    std::string password )
```

Check whether login was successful. Return error if not.

## Parameters

<i>username</i>	
<i>password</i>	

## Returns

error or empty string

**3.37.3.3 DoLogout()**

```
bool UserManager::DoLogout (
    const char * ptr )
```

Logout user. try to get user from cookie. Logout user, reset controller update interval and erase cookie. If user does not exist, try only to delete cookie from map.

## Parameters

in	<i>cookie.</i>	
----	----------------	--

## Returns

false if cookie does not exist in map.

**3.37.3.4 DoRegistration()**

```
std::string UserManager::DoRegistration (
    std::string username,
```

```
std::string pw1,  
std::string pw2 )
```

Check whether registration was successful. Return error or empty string.

## Parameters

<i>username</i>	
<i>pw1</i>	
<i>pw2</i>	

## Returns

error code or empty string.

**3.37.3.5 GenerateCookie()**

```
std::string UserManager::GenerateCookie (
    std::string username )
```

Create random 32 characters to generates cookie. And maps cookie and given user.

## Parameters

in	<i>username</i>	(username which is mapped on cookie)
----	-----------------	--------------------------------------

## Returns

returns cookie as string.

**3.37.3.6 GetAllWorlds()**

```
nlohmann::json UserManager::GetAllWorlds (
    std::string username ) const
```

Get all worlds. Array of jsons returned with values: user, name (of world) and port.

## Parameters

in	<i>user</i>	to create list for. return json
----	-------------	---------------------------------

**3.37.3.7 GetNextPort()**

```
int UserManager::GetNextPort ( )
```

Returns next port.

### 3.37.3.8 GetSharedWorlds()

```
nlohmann::json UserManager::GetSharedWorlds (
    std::string username ) const
```

Get all shared worlds of a user. Array of jsons returned with values: user, name (of world) and port.

#### Parameters

in	<i>user</i>	to create list for. return json
----	-------------	---------------------------------

### 3.37.3.9 GetUser()

```
User * UserManager::GetUser (
    std::string username ) const
```

Returns given user, if exists

#### Parameters

in	<i>username</i>	
----	-----------------	--

#### Returns

user, if exists, nullptr otherwise

### 3.37.3.10 GetUserFromCookie()

```
std::string UserManager::GetUserFromCookie (
    const char * ptr ) const
```

Get user from cookie.

#### Parameters

in	<i>resp</i>	(reference to response)
----	-------------	-------------------------

#### Returns

username, "\$no\_cookie" or "\$no\_user"

### 3.37.3.11 GrantAccessTo()

```
int UserManager::GrantAccessTo (
    std::string user1,
    std::string user2,
    std::string world )
```

Grant other user acces to a own world.

#### Parameters

in	<i>user1</i>	(user which gives access world to)
in	<i>user2</i>	(user to give access to)
in	<i>world</i>	(world which to grant access to)

#### Returns

error\_code

The documentation for this class was generated from the following files:

- /home/fux/Documents/programming/Der-Zug/src/gui/src/users/user\_manager.h
- /home/fux/Documents/programming/Der-Zug/src/gui/src/users/user\_manager.cc

## 3.38 World Class Reference

```
#include <world.h>
```

### Public Member Functions

- [World](#) (std::string base\_path, std::string path, int port)
- [~World](#) ()
- int [port](#) ()
- nllohmann::json [GetPage](#) (std::string path)

### 3.38.1 Detailed Description

#### Author

fux Class storing all pages and information of a world.

### 3.38.2 Constructor & Destructor Documentation

#### 3.38.2.1 World()

```
World::World (
    std::string base_path,
    std::string path,
    int port )
```

Constructor generating all pages.

**Parameters**

in	<i>base_path</i>	to user-directory.
----	------------------	--------------------

**3.38.2.2 ~World()**

```
World::~~World ( )
```

Destructor deleteing all pages.

**3.38.3 Member Function Documentation****3.38.3.1 GetPage()**

```
nlohmann::json World::GetPage (
    std::string path )
```

Calls GetPage value of referenced page and add short paths to data-json.

**Parameters**

in	<i>path</i>	to category/area/object.
----	-------------	--------------------------

**Returns**

json with page-data, path to template and short-paths.

The documentation for this class was generated from the following files:

- /home/fux/Documents/programming/Der-Zug/src/gui/src/world/world.h
- /home/fux/Documents/programming/Der-Zug/src/gui/src/world/world.cc

**3.39 Worlds Class Reference**

```
#include <worlds.h>
```

**Public Member Functions**

- [Worlds](#) (std::string base\_path, int start\_port)
- [~Worlds](#) ()
- std::string [GetPage](#) (std::string path)



### 3.39.1 Detailed Description

#### Author

flux Class storing all existing worlds by it's full path. Full path means: [base\_path]/[user]/files/[world] Apart from storing, the class redirects all requests to the matching world and assigns port to every world, to potentially run on.

### 3.39.2 Constructor & Destructor Documentation

#### 3.39.2.1 Worlds()

```
Worlds::Worlds (
    std::string base_path,
    int start_port )
```

Constructor, loading all worlds.

#### Parameters

in	<i>base_path</i>	to user-directories.
in	<i>start_port</i>	from which to increment and assign ports.

#### 3.39.2.2 ~Worlds()

```
Worlds::~~Worlds ( )
```

Destructor deleteing all worlds.

### 3.39.3 Member Function Documentation

#### 3.39.3.1 GetPage()

```
std::string Worlds::GetPage (
    std::string path )
```

Seves page of requested category/ object. Acctually get json-data and path to tempate, then calls ParseTemplate().

#### Parameters

in	<i>path</i>	(url)
----	-------------	-------

**Returns**

rendered page.

The documentation for this class was generated from the following files:

- /home/fux/Documents/programming/Der-Zug/src/gui/src/world/worlds.h
- /home/fux/Documents/programming/Der-Zug/src/gui/src/world/worlds.cc

# Index

- ~World
  - World, [50](#)
- ~Worlds
  - Worlds, [51](#)
- AddElem
  - ServerFrame, [25](#)
- AddFile
  - User, [36](#)
- AddLocation
  - User, [36](#)
- AddNewObject
  - User, [37](#)
- AddRequest
  - User, [37](#)
- Area, [5](#)
  - CreatePageData, [5](#)
- Backups
  - ServerFrame, [25](#)
- Category, [7](#)
  - CreatePageData, [8](#)
- CheckAccessToLocations
  - User, [37](#)
- CheckGameRunning
  - User, [38](#)
- CheckRunning
  - ServerFrame, [26](#)
- child\_nodes\_
  - Page, [21](#)
- CreateBackup
  - User, [38](#)
- CreateNewWorld
  - User, [38](#)
- CreatePageData
  - Area, [5](#)
  - Category, [8](#)
  - Page, [20](#)
- CreateRequest
  - ServerFrame, [26](#)
- DelElem
  - ServerFrame, [26](#)
- DeleteBackup
  - User, [39](#)
- DeleteFile
  - User, [39](#)
- DeleteObject
  - User, [40](#)
- DeleteUser
  - UserManager, [44](#)
- DeleteWorld
  - User, [40](#)
- DelUser
  - ServerFrame, [26](#)
- DoLogin
  - ServerFrame, [27](#)
  - UserManager, [45](#)
- DoLogout
  - ServerFrame, [27](#)
  - UserManager, [45](#)
- DoRegistration
  - ServerFrame, [27](#)
  - UserManager, [45](#)
- GenerateChildNodes
  - Page, [21](#)
- GenerateCookie
  - UserManager, [47](#)
- GenerateParentNodes
  - Page, [21](#)
- GetAllWorlds
  - UserManager, [47](#)
- GetBackups
  - User, [40](#)
- GetCategory
  - User, [40](#)
- GetLog
  - ServerFrame, [28](#)
- GetNextPort
  - UserManager, [47](#)
- GetObjects
  - User, [41](#)
- GetOverview
  - User, [41](#)
- GetPage
  - World, [50](#)
  - Worlds, [51](#)
- GetPortOfWorld
  - User, [41](#)
- GetSharedWorlds
  - UserManager, [47](#)
- GetUser
  - UserManager, [48](#)
- GetUserFromCookie
  - UserManager, [48](#)
- GetWorld
  - User, [42](#)
- GrantAccessTo

- ServerFrame, [28](#)
  - UserManager, [48](#)
- include\_template
  - inja::Environment, [9](#)
- inja::AstNode, [6](#)
- inja::BlockNode, [7](#)
- inja::Environment, [8](#)
  - include\_template, [9](#)
- inja::ExpressionListNode, [10](#)
- inja::ExpressionNode, [10](#)
- inja::FileError, [11](#)
- inja::ForArrayStatementNode, [11](#)
- inja::ForObjectStatementNode, [12](#)
- inja::ForStatementNode, [12](#)
- inja::FunctionNode, [13](#)
- inja::FunctionStorage, [14](#)
- inja::FunctionStorage::FunctionData, [13](#)
- inja::IfStatementNode, [15](#)
- inja::IncludeStatementNode, [15](#)
- inja::InjaError, [16](#)
- inja::JsonError, [17](#)
- inja::JsonNode, [17](#)
- inja::Lexer, [18](#)
- inja::LiteralNode, [18](#)
- inja::NodeVisitor, [19](#)
- inja::Parser, [22](#)
- inja::ParserError, [22](#)
- inja::Renderer, [23](#)
- inja::RenderError, [23](#)
- inja::SetStatementNode, [30](#)
- inja::SourceLocation, [31](#)
- inja::StatementNode, [31](#)
- inja::StatisticsVisitor, [32](#)
- inja::Template, [32](#)
- inja::TextNode, [33](#)
- inja::Token, [33](#)
- IsRunning
  - ServerFrame, [28](#)
- LoginPage
  - ServerFrame, [29](#)
- Page, [20](#)
  - child\_nodes\_, [21](#)
  - CreatePageData, [20](#)
  - GenerateChildNodes, [21](#)
  - GenerateParentNodes, [21](#)
  - parent\_nodes\_, [21](#)
- parent\_nodes\_
  - Page, [21](#)
- RemoveRequest
  - User, [42](#)
- RestoreBackup
  - User, [42](#)
- SafeUser
  - User, [43](#)
- ServeFile
  - ServerFrame, [29](#)
- ServerFrame, [24](#)
  - AddElem, [25](#)
  - Backups, [25](#)
  - CheckRunning, [26](#)
  - CreateRequest, [26](#)
  - DelElem, [26](#)
  - DelUser, [26](#)
  - DoLogin, [27](#)
  - DoLogout, [27](#)
  - DoRegistration, [27](#)
  - GetLog, [28](#)
  - GrantAccessTo, [28](#)
  - IsRunning, [28](#)
  - LoginPage, [29](#)
  - ServeFile, [29](#)
  - ServerFrame, [25](#)
  - Start, [29](#)
  - StartGame, [30](#)
  - WriteObject, [30](#)
- Start
  - ServerFrame, [29](#)
- StartGame
  - ServerFrame, [30](#)
- User, [34](#)
  - AddFile, [36](#)
  - AddLocation, [36](#)
  - AddNewObject, [37](#)
  - AddRequest, [37](#)
  - CheckAccessToLocations, [37](#)
  - CheckGameRunning, [38](#)
  - CreateBackup, [38](#)
  - CreateNewWorld, [38](#)
  - DeleteBackup, [39](#)
  - DeleteFile, [39](#)
  - DeleteObject, [40](#)
  - DeleteWorld, [40](#)
  - GetBackups, [40](#)
  - GetCategory, [40](#)
  - GetObjects, [41](#)
  - GetOverview, [41](#)
  - GetPortOfWorld, [41](#)
  - GetWorld, [42](#)
  - RemoveRequest, [42](#)
  - RestoreBackup, [42](#)
  - SafeUser, [43](#)
  - User, [35](#)
  - WriteObject, [43](#)
- userManager, [43](#)
  - DeleteUser, [44](#)
  - DoLogin, [45](#)
  - DoLogout, [45](#)
  - DoRegistration, [45](#)
  - GenerateCookie, [47](#)
  - GetAllWorlds, [47](#)
  - GetNextPort, [47](#)
  - GetSharedWorlds, [47](#)

- GetUser, [48](#)
- GetUserFromCookie, [48](#)
- GrantAccessTo, [48](#)
- UserManager, [44](#)

World, [49](#)

- ~World, [50](#)
- GetPage, [50](#)
- World, [49](#)

Worlds, [50](#)

- ~Worlds, [51](#)
- GetPage, [51](#)
- Worlds, [51](#)

WriteObject

- ServerFrame, [30](#)
- User, [43](#)