GameCreator

Generated by Doxygen 1.8.20

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Chapter 2

Class Index

2.1 Class List

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

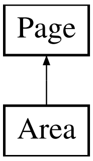
Chapter 3

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

fux

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 opject.

Parameters

`in`	*path*	to area or object.

Returns

json with information.

Reimplemented from Page.

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

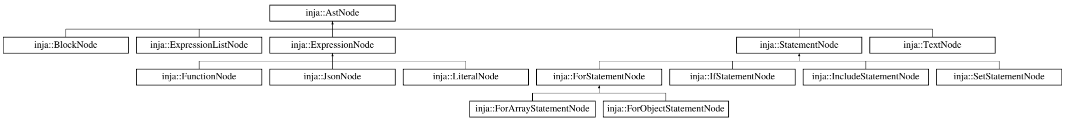
- /home/fux/Documents/programming/Der-Zug/src/gui/src/page/area.h
- /home/fux/Documents/programming/Der-Zug/src/gui/src/page/area.cc

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

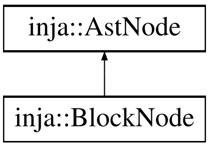
Base node class for the abstract syntax tree (AST).

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**

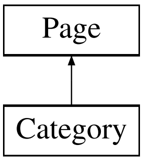
The documentation for this class was generated from the following file:

- /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	*path*	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]
```

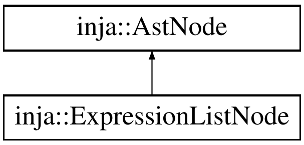
Includes a template with a given name into the environment. Then, a template can be rendered in another template using the include "<name>" syntax.

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

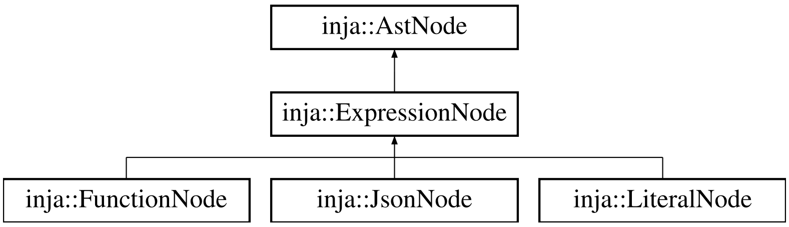
- std::vector< std::shared_ptr< ExpressionNode > > **rpn_output**

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

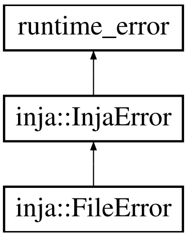
Additional Inherited Members

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)

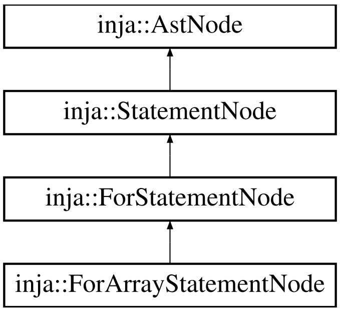
Additional Inherited Members

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

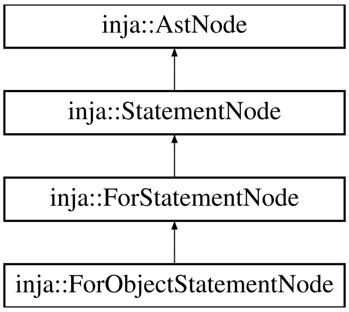
- const std::string **value**

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

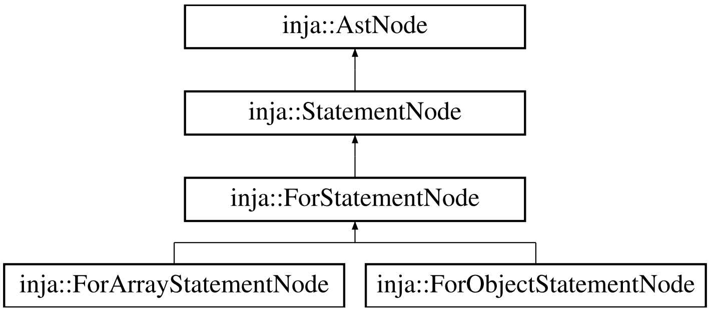
- const std::string **key**
- const std::string **value**

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

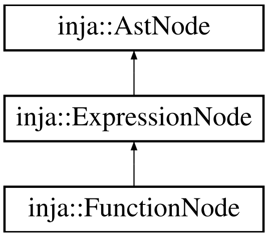
- const Operation **operation**
- const CallbackFunction **callback**

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

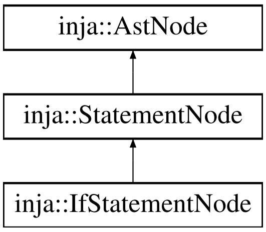
Class for builtin functions and user-defined callbacks.

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

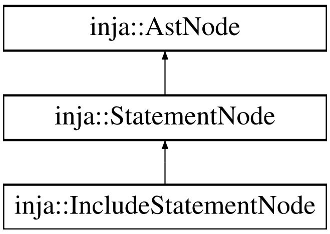
- ExpressionListNode **condition**
- BlockNode **true_statement**
- BlockNode **false_statement**
- BlockNode ∗const **parent**
- const bool **is_nested**
- bool **has_false_statement** {false}

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

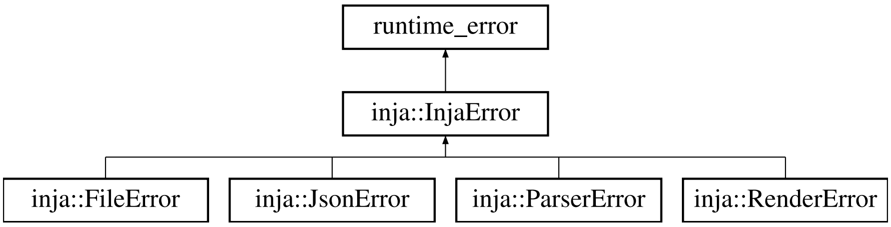
- const std::string **file**

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

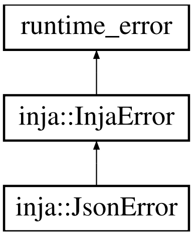
- const std::string **type**
- const std::string **message**
- const SourceLocation **location**

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)

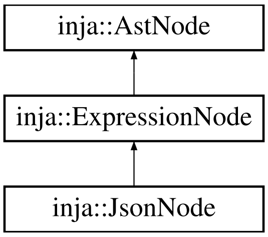
Additional Inherited Members

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

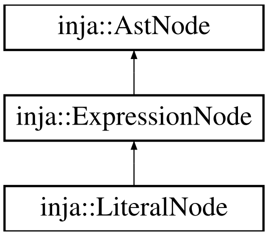
Class for lexing an inja Template.

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

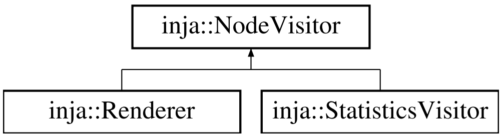
- const nlohmann::json **value**

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

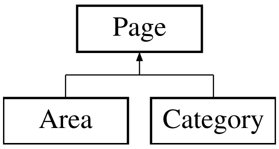
The documentation for this class was generated from the following file:

- /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

fux Base class for pages. Class is abstract and never instaciated. 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 templae. Used by category and area, however not for objects.

Parameters

`in`	*path*	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

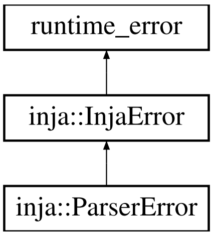
Class for parsing an inja Template.

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:

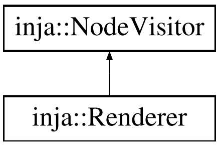
- /home/fux/Documents/programming/Der-Zug/src/gui/src/users/inja.hpp

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

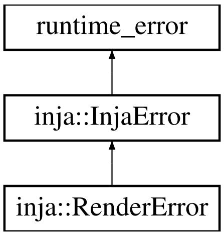
Class for rendering a Template with data.

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 httplib::Request &req, httplib::Response &resp) const
- void DoLogin (const httplib::Request &req, httplib::Response &resp)
- void DoRegistration (const httplib::Request &req, httplib::Response &resp)
- void DoLogout (const httplib::Request &req, httplib::Response &resp)
- void DelUser (const httplib::Request &req, httplib::Response &resp)

  Deletes a user. User is deleted on own request.
- void ServeFile (const httplib::Request &req, httplib::Response &resp, bool backup=false) const
- void AddElem (const httplib::Request &req, httplib::Response &resp)

  Add new element (world, subcategory, object).
- void DelElem (const httplib::Request &req, httplib::Response &resp)

  Delete element (world, subcategory, object).
- void WriteObject (const httplib::Request &req, httplib::Response &resp)

  Write (modified) object.
- void Backups (const httplib::Request &req, httplib::Response &resp, std::string action)

  Creates a backup of the selected world.
- void GrantAccessTo (const httplib::Request &req, httplib::Response &resp)

  Grants given user access to world of other user.
- void CreateRequest (const httplib::Request &req, httplib::Response &resp)

  Grants given user access to world of other user.
- void CheckRunning (const httplib::Request &req, httplib::Response &resp)

  Checks whether game is still running.
- void GetLog (const httplib::Request &req, httplib::Response &resp)

  Checks whether game is still running.
- void StartGame (const httplib::Request &req, httplib::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

fux

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 httplib::Request & req,
            httplib::Response & resp )
```

Add new element (world, subcategory, object).

Parameters

in	*req*	(reference to request)
in,out	*resp*	(reference to response)

3.28.3.2 Backups()

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

Creates a backup of the selected world.

Parameters

in	*req*	(reference to request)
in,out	*resp*	(reference to response)

3.28.3.3 CheckRunning()

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

Checks whether game is still running.

Parameters

in	*req*	(reference to request)
in,out	*resp*	(reference to response)

3.28.3.4 CreateRequest()

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

Grants given user access to world of other user.

Parameters

in	*req*	(reference to request)
in,out	*resp*	(reference to response)

3.28.3.5 DelElem()

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

Delete element (world, subcategory, object).

Parameters

in	*req*	(reference to request)
in,out	*resp*	(reference to response)

3.28.3.6 DelUser()

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

Deletes a user. User is deleted on own request.

Parameters

in	*req*	(reference to request)
in,out	*resp*	(reference to response)

3.28.3.7 DoLogin()

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

function to handle login

Parameters

in	*req*	(reference to request)
in,out	*resp*	(server response)
in	*user_manager*	(user_manager to check whether user exists)

3.28.3.8 DoLogout()

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

logout a user

Parameters

in	*req*	(reference to request)
in,out	*resp*	(server response)

3.28.3.9 DoRegistration()

```
void ServerFrame::DoRegistration (
            const httplib::Request & req,
            httplib::Response & resp )
```

function to handle registration

Parameters

in	*req*	(reference to request)
in,out	*resp*	(server response)

3.28.3.10 GetLog()

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

Checks whether game is still running.

Parameters

in	*req*	(reference to request)
in,out	*resp*	(reference to response)

3.28.3.11 GrantAccessTo()

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

Grants given user access to world of other user.

Parameters

in	*req*	(reference to request)
in,out	*resp*	(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 httplib::Request & req,
            httplib::Response & resp ) const
```

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

Parameters

in	*req*	(reference to request)
in,out	*resp*	(server response)

3.28.3.14 ServeFile()

```
void ServerFrame::ServeFile (
            const httplib::Request & req,
            httplib::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	*req*	(reference to request)
in,out	*resp*	(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	*port*	(Port to start server on)

3.28.3.16 StartGame()

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

Checks whether game is still running.

Parameters

`in`	*req*	(reference to request)
`in,out`	*resp*	(reference to response)

3.28.3.17 WriteObject()

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

Write (modified) object.

Parameters

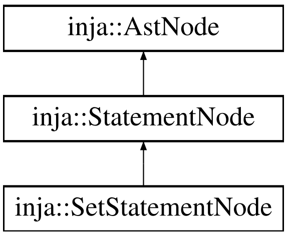
`in`	*req*	(reference to request)
`in,out`	*resp*	(reference to response)

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

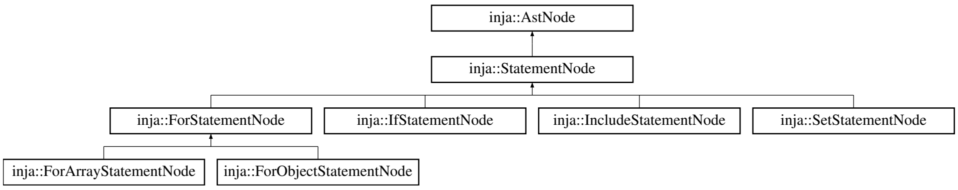
- size_t **line**
- size_t **column**

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:

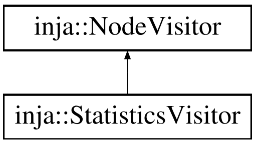
- /home/fux/Documents/programming/Der-Zug/src/gui/src/users/inja.hpp

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

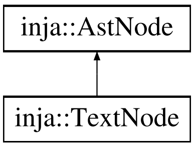
The main inja Template.

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

fux

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	*username.*	
in	*pw*	(password).
in	*path*	(path to folders).
in	*categories*	(all currently available categories)

Author

fux

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	*username.*	
in	*pw*	(password).
in	*path*	(path to folders).
in	*locations*	(locations available for this user).
in	*categories*	(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

path	
name	

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	*user*	(user who granted access)
in	*world*	(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

path	
id	
force	

Returns

ErrorCode.

3.36.3.4 AddRequest()

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

Adds a new access-request.

Parameters

in	*user*	(user who asks for access)
in	*world*	(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	*path.*	

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	*path*	(path to extract information from)
in	*number*	(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	*user*	
in	*world*	

Returns

ErrorCode.

3.36.3.8 CreateNewWorld()

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

Creates a new world for this user.

Parameters

name	(Name of the world)
port	

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	*user*	
in	*backup*	

Returns

ErrorCode.

3.36.3.10 DeleteFile()

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

delete a given file (subcategory)

Parameters

in	*path*	(path to category)
in	*subcategory*	return ErrorCode.

3.36.3.11 DeleteObject()

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

delete a given object

Parameters

in	*path*	(path to json)
in	*object*	(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	*world*	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	*user*	
in	*world*	

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	*shared_worlds*	(json with all shared_worlds and ports)
in	*all_worlds*	(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	*path*	(path to world)
in	*attr*	

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

path	(path to current world)
world	(name of accessed world)
port	(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	*user*	(user who asks for access)
in	*world*	(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	*user*	
in	*backup*	(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	*request*	(request to create new, or change existsing)
in	*force*	(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

fux class storing all users. TODO (fux): 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

fux

3.37.3 Member Function Documentation

3.37.3.1 DeleteUser()

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

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

Parameters

in	*username*	

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

username	
password	

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	*cookie.*	

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

username	
pw1	
pw2	

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	*username*	(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	*user*	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	*user*	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	*username*	

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	*resp*	(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	*user1*	(user which gives access world to)
in	*user2*	(user to give access to)
in	*world*	(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** ()
- nlohmann::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	*base_path*	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	*path*	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

fux 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 assignes 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	*base_path*	to user-directories.
in	*start_port*	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	*path*	(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