Web framework C++

Generated by Doxygen 1.8.15

Contents

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

File Index

3.1 File List

Here is a list of all files with brief descriptions:

Chapter 4

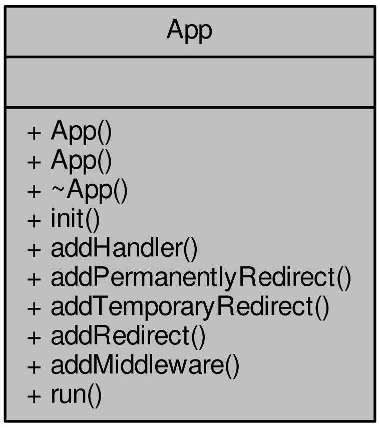
Class Documentation

4.1 App Class Reference

The main class of the framework. Each object of this class is an independent web-application, which could be configured by handlers, middleware etc.

```
#include <app.h>
```

Collaboration diagram for App:



Public Member Functions

- App (std::string &ip, int port=80, bool isIPv6=false, const char ∗logFilePath=nullptr)
- App (InitParams &params)
- ∼App ()
- bool init ()
- void addHandler (Handler ∗handler)
- void addPermanentlyRedirect (const char ∗uri, const char ∗target)
- void addTemporaryRedirect (const char ∗uri, const char ∗target)
- void addRedirect (const char ∗uri, const char ∗target, int code)
- void addMiddleware (Middleware ∗middleware)
- void run ()

4.1.1 Detailed Description

The main class of the framework. Each object of this class is an independent web-application, which could be configured by handlers, middleware etc.

This class implements web-application, which is running on given and port. It supports IPv6 and can capture log in the file if given. Use Handlers, Middleware and set Redirects to adjust it.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 App() [1/2]

```
App::App (
            std::string & ip,
            int port = 80,
            bool isIPv6 = false,
            const char * logFilePath = nullptr )  [explicit]
```

Create a new web application, which is only adjusted to ip address. You can init this and add some handlers etc. to run this

Parameters

ip	text representation of ip address, like 127.0.0.1 or 0:0:0:0:0:0:0:1 (if IPv6)
port	port in range [0, 65535]
isIPv6	set true, if param ip is version 6
logFilePath	if you want to create log file, give a file path, or null otherwise

4.1.2.2 App() [2/2]

```
App::App (
            InitParams & params )  [explicit]
```

Create a new web application, by command line arguments using InitParams object

Parameters

params	Give an object params, which was created by InitParams class from command line arguments

4.1.2.3 ∼App()

```
App::∼App ( )
```

Destructor delete all added handlers and all middleware

4.1.3 Member Function Documentation

4.1.3.1 addHandler()

```
void App::addHandler (
            Handler * handler )
```

To configure your application create and add some handlers

Parameters

handler	object of class Handler (could be inherited) with overridden function exec

4.1.3.2 addMiddleware()

```
void App::addMiddleware (
            Middleware * middleware )
```

Add object of class Middleware, which has got overridden function exec to do given operations on every request. All handlers could access to any middleware and perform adjusted actions.

Parameters

middleware	object of class Middleware (could be inherited)

4.1.3.3 addPermanentlyRedirect()

```
void App::addPermanentlyRedirect (
            const char * uri,
            const char * target )
```

Add redirection, which is meant to last forever. The original URL should not be used anymore and that the new one is preferred. Search engine robots trigger an update of the associated URL for the resource in their indexes. (HTTP code 301)

Parameters

uri	original uri path, which is deprecated (outdated)
target	new uri address of mentioned page

4.1.3.4 addRedirect()

```
void App::addRedirect (
            const char * uri,
            const char * target,
            int code )
```

To adjust any redirection using status code. For example, 304 (Not Modified) redirects a page to the locally cached copy, and 300 (Multiple Choice) is a manual redirection: the body, presented by the browser as a Web page, lists the possible redirection and the user clicks on one to select it.

Parameters

uri	original uri path, which is deprecated (outdated)
target	new uri address of mentioned page
code	HTTP code redirection status of response

4.1.3.5 addTemporaryRedirect()

```
void App::addTemporaryRedirect (
            const char * uri,
            const char * target )
```

Temporary redirect can be used, if for some time the requested resource cannot be accessed from its canonical location, but it can be accessed from another place. Search engine robots don't memorize the new, temporary link. Temporary redirection are also used when creating, updating and deleting resources to present temporary progress pages.

Parameters

uri	original uri path, which is deprecated (outdated)
target	new uri address of mentioned page

4.1.3.6 init()

```
bool App::init ( )
```

Use this function to open socket for listening on declared ip address and port. After creating on object you should use this function to startup web-socket

Returns

true if ip address and port were valid and available, false - otherwise, please, use another address to continue

4.1.3.7 run()

```
void App::run ( )
```

Start listening for request. This method startup the system, where on every request from clients all added handlers and middleware create a response and send it to client.

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

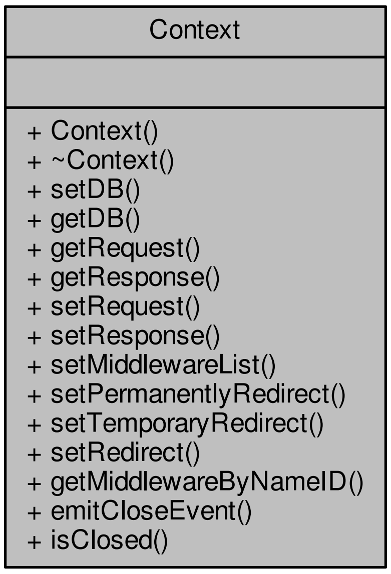
- /home/a_krava/projects/progbase3/web_server/include/app.h
- /home/a_krava/projects/progbase3/web_server/src/app.cpp

4.2 Context Class Reference

This class is wrapper for important data (like Response, DB, Middleware etc.), which is needed to handlers.

```
#include <context.h>
```

Collaboration diagram for Context:



Public Member Functions

- Context ()
- ∼Context ()
- void setDB (DBManager ∗db)
- DBManager ∗ getDB ()
- Request ∗ getRequest ()
- Response ∗ getResponse ()
- void setRequest (Request ∗request)
- void setResponse (Response ∗response)
- void setMiddlewareList (std::vector< Middleware ∗> ∗middlewareList)
- void setPermanentlyRedirect (const char ∗uri)
- void setTemporaryRedirect (const char ∗uri)
- void setRedirect (const char ∗uri, int code)
- Middleware ∗ getMiddlewareByNameID (const char ∗nameID)
- void emitCloseEvent ()
- bool isClosed ()

4.2.1 Detailed Description

This class is wrapper for important data (like Response, DB, Middleware etc.), which is needed to handlers.

This class collects a data about current request, which was parsed from str, have a pointer to Response object, which will be serialized to client in future (here could be some written data from previous responses), also there are references to all added middleware (you could get some by id) and database, which is ready to perform method exec

4.2.2 Constructor & Destructor Documentation

4.2.2.1 Context()

```
Context::Context ( )
```

Constructor create an object of this class: creating Request and Response objects, and setting NULL to db and middlewareList

4.2.2.2 ∼Context()

```
Context::∼Context ( )
```

Deleting Request, Response and DB objects, if they are not NULL

4.2.3 Member Function Documentation

4.2.3.1 emitCloseEvent()

```
void Context::emitCloseEvent ( )
```

Emit signal to stop executing operation. Handler, which used this will be last executed handler in app

4.2.3.2 getDB()

```
DBManager * Context::getDB ( )
```

Returns db, if wasn't set - nullptr

Returns

Object of class DBManager

4.2.3.3 getMiddlewareByNameID()

```
Middleware * Context::getMiddlewareByNameID (
             const char * nameID )
```

Method returns added Middleware by id (in string)

Parameters

nameID	id of middleware, which was set at startup

Returns

object of Middleware (could be inherited)

4.2.3.4 getRequest()

```
Request * Context::getRequest ( )
```

Gives current request

Returns

object of Request class

4.2.3.5 getResponse()

```
Response * Context::getResponse ( )
```

Gives current response. Could be modified by previous handlers

Returns

object of Response class

4.2.3.6 isClosed()

```
bool Context::isClosed ( )
```

Checks, if handlers emitted CloseEvent

Returns

true, if there were emitted close event, false otherwise

4.2.3.7 setDB()

```
void Context::setDB (
            DBManager * db )
```

Method sets the object of database, created by DBManager. All handlers can access it

Parameters

db	Object of class DBManager

4.2.3.8 setMiddlewareList()

```
void Context::setMiddlewareList (
            std::vector< Middleware *> * middlewareList )
```

sets vector of Middleware objects, which can be accessed by handlers

Parameters

middlewareList	std::vector of Middleware objects

4.2.3.9 setPermanentlyRedirect()

```
void Context::setPermanentlyRedirect (
             const char * uri )
```

Set permanent (code 301) redirect headers to Response

Parameters

uri	destination uri, where current request will be redirected

4.2.3.10 setRedirect()

```
void Context::setRedirect (
             const char * uri,
             int code )
```

Set redirect headers to Response

Parameters

uri	destination uri, where current request will be redirected
code	http code of redirect

4.2.3.11 setRequest()

```
void Context::setRequest (
             Request * request )
```

Deleting existing Request and setting new one

Parameters

request	object of Request class (could be inherited)

4.2.3.12 setResponse()

```
void Context::setResponse (
             Response * response )
```

Deleting existing Response and setting new one

Parameters

response	object of Response class (could be inherited)

4.2.3.13 setTemporaryRedirect()

```
void Context::setTemporaryRedirect (
            const char * uri )
```

Set temporary (code 302) redirect headers to Response

Parameters

uri	destination uri, where current request will be redirected

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

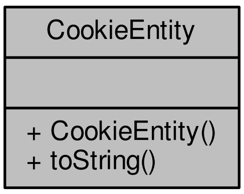
- /home/a_krava/projects/progbase3/web_server/include/context.h
- /home/a_krava/projects/progbase3/web_server/src/context.cpp

4.3 CookieEntity Class Reference

Class wrapper for Cookies. Allow you adjust parameters od each http cookie. Used by CookieMiddleware.

```
#include <cookie_entity.h>
```

Collaboration diagram for CookieEntity:



Public Member Functions

- CookieEntity (const char *value, time_t expires=-1, size_t maxAge_sec=std::string::npos, const char *domain=nullptr, const char *path=nullptr, bool httpOnly=false)
- std::string toString ()

4.3.1 Detailed Description

Class wrapper for Cookies. Allow you adjust parameters od each http cookie. Used by CookieMiddleware.

Object of this class consist of key-value pair, and some options for it, like date expires, max age, domain, path, option http only

4.3.2 Constructor & Destructor Documentation

4.3.2.1 CookieEntity()

```
CookieEntity::CookieEntity (
             const char * value,
             time_t expires = -1,
             size_t maxAge_sec = std::string::npos,
             const char * domain = nullptr,
             const char * path = nullptr,
             bool httpOnly = false )
```

Constructs a cookie entity with parameters

Parameters

value	value of cookie
expires	the maximum lifetime of the cookie as time_t
maxAge_sec	number of seconds until the cookie expires.
domain	specifies those hosts to which the cookie will be sent.
path	indicates a URL path that must exist in the requested resource before sending the Cookie header
httpOnly	HTTP-only cookies aren't accessible via JavaScript

4.3.3 Member Function Documentation

4.3.3.1 toString()

```
std::string CookieEntity::toString ( )
```

Method is used to serialize itself

Returns

serialized string like: "<cookie-value>; Expires=<date>; Max-Age=<non-zero-digit> ..."

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

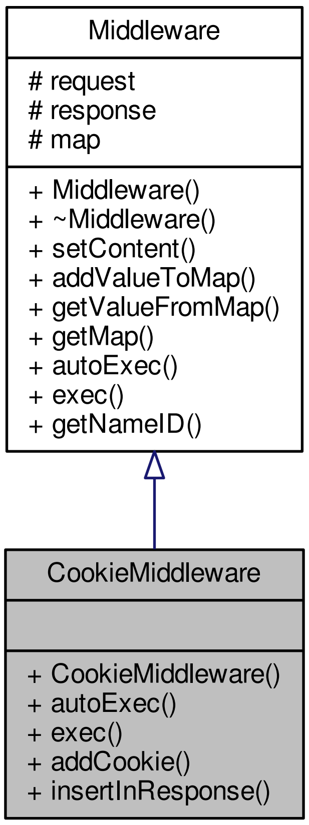
- /home/a_krava/projects/progbase3/web_server/include/cookie_entity.h
- /home/a_krava/projects/progbase3/web_server/src/cookie_entity.cpp

4.4 CookieMiddleware Class Reference

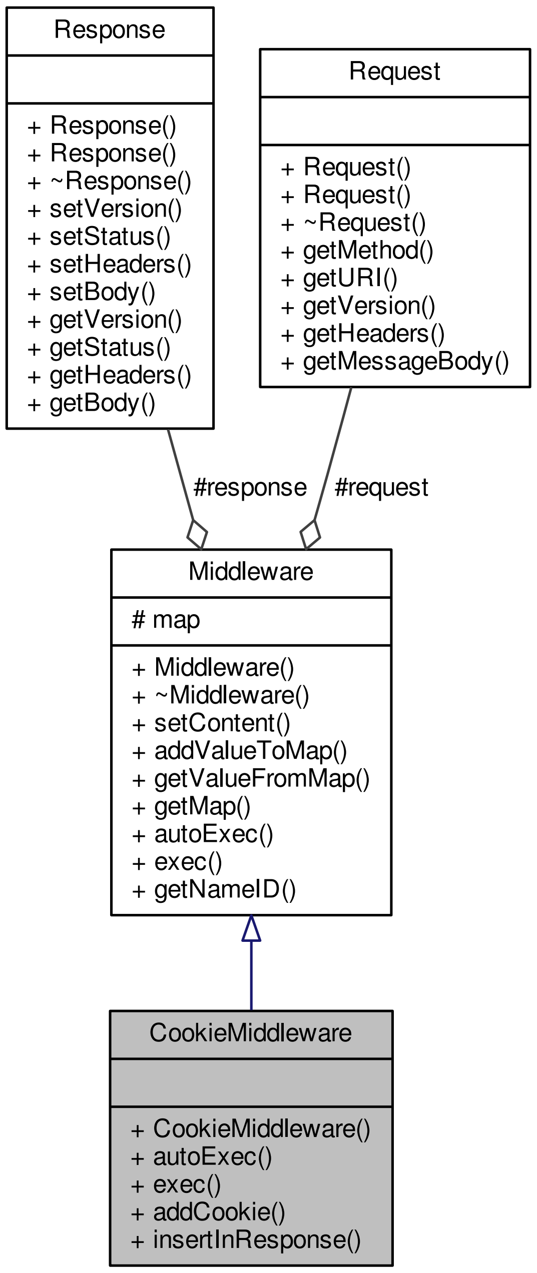
inherited class to parse cookie from http request

```
#include <cookie_middleware.h>
```

Inheritance diagram for CookieMiddleware:



Collaboration diagram for CookieMiddleware:



Public Member Functions

- CookieMiddleware (const char ∗nameID)
- bool autoExec ()
- void exec ()
- void addCookie (const char ∗key, CookieEntity &value)
- void insertInResponse ()

Additional Inherited Members

4.4.1 Detailed Description

inherited class to parse cookie from http request

CookieMiddleware is intended to parse cookie from http request, fill response with cookies

4.4.2 Constructor & Destructor Documentation

4.4.2.1 CookieMiddleware()

```
CookieMiddleware::CookieMiddleware (
             const char * nameID )
```

create middleware

Parameters

nameID	name id

4.4.3 Member Function Documentation

4.4.3.1 addCookie()

```
void CookieMiddleware::addCookie (
             const char * key,
             CookieEntity & value )
```

add CookieEntity to response cookies

Parameters

key	key for entity
value	CookieEntity object

4.4.3.2 autoExec()

```
bool CookieMiddleware::autoExec ( ) [virtual]
```

Check if there are cookie in request

Returns

true, if there are cookie in request

Implements Middleware.

4.4.3.3 exec()

```
void CookieMiddleware::exec ( )  [virtual]
```

parse cookies from http request

Implements Middleware.

4.4.3.4 insertInResponse()

```
void CookieMiddleware::insertInResponse ( )
```

set response cookies in response headers

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

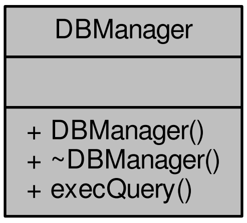
- /home/a_krava/projects/progbase3/web_server/include/cookie_middleware.h
- /home/a_krava/projects/progbase3/web_server/src/cookie_middleware.cpp

4.5 DBManager Class Reference

allow perform sql queries to db

```
#include <db_manager.h>
```

Collaboration diagram for DBManager:



Public Member Functions

- DBManager (const char ∗filePath)
- ∼DBManager ()
- bool execQuery (const char ∗statement, std::vector< std::vector< std::string >> &result_vec, char ∗data[ ], int num)

4.5.1 Detailed Description

allow perform sql queries to db

Realisation of wrapper for SQLite database

4.5.2 Constructor & Destructor Documentation

4.5.2.1 DBManager()

```
DBManager::DBManager (
            const char ∗ filePath )
```

Opening database

Parameters

filePath	file path to database

4.5.2.2 ∼DBManager()

```
DBManager::∼DBManager ( )
```

Closing opened connection to db

4.5.3 Member Function Documentation

4.5.3.1 execQuery()

```
bool DBManager::execQuery (
            const char ∗ statement,
            std::vector< std::vector< std::string >> & result_vec,
            char ∗ data[ ],
            int num )
```

Execute SQL statement, with binding values by escaping with '?' in statement and including them in data array

Parameters

statement	SQL query
result_vec	out parameter, used to write result table of executed query
data	some data can be missed by ? in statement. So it's providing in this data array
num	number of binding (preparing) values

Returns

true if executed successfully, false otherwise

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

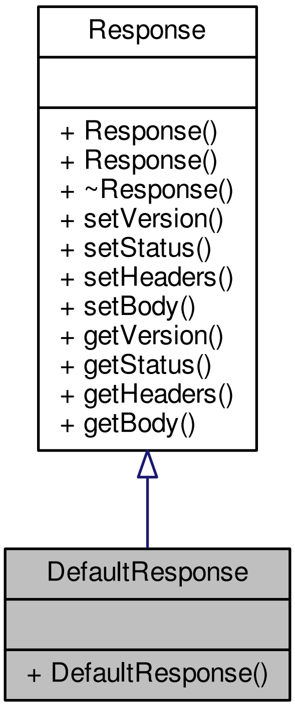
- /home/a_krava/projects/progbase3/web_server/include/db_manager.h
- /home/a_krava/projects/progbase3/web_server/src/db_manager.cpp

4.6 DefaultResponse Class Reference

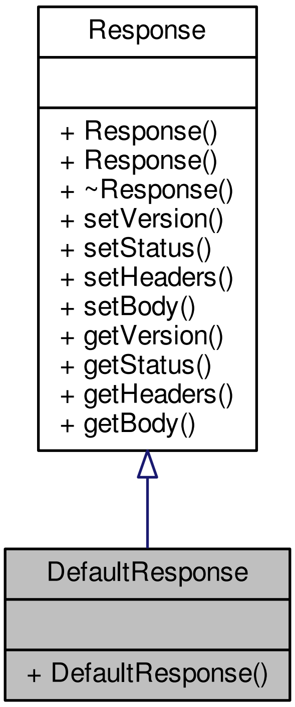
Response class which is intended to make sample html pages on status codes.

```
#include <default_response.h>
```

Inheritance diagram for DefaultResponse:



Collaboration diagram for DefaultResponse:



Public Member Functions

- DefaultResponse (int status_code, const char ∗body=nullptr)

4.6.1 Detailed Description

Response class which is intended to make sample html pages on status codes.

Inherited class DefaultResponse from Response for setting stubs for non-realized functionality

4.6.2 Constructor & Destructor Documentation

4.6.2.1 DefaultResponse()

```
DefaultResponse::DefaultResponse (
             int status_code,
             const char * body = nullptr )
```

Create DefaultResponse with code status or custom body

Parameters

status_code	http code status, set the body for its reason phrase, if status_code $<$ 0, set the body from param body
body	custom body page

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

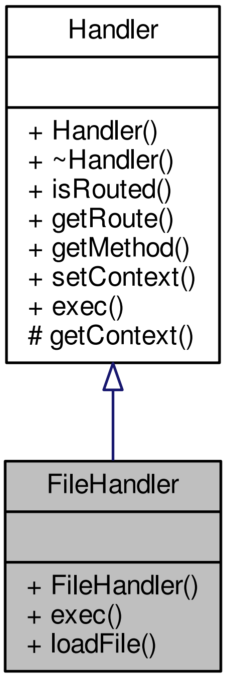
- /home/a_krava/projects/progbase3/web_server/include/default_response.h
- /home/a_krava/projects/progbase3/web_server/src/default_response.cpp

4.7 FileHandler Class Reference

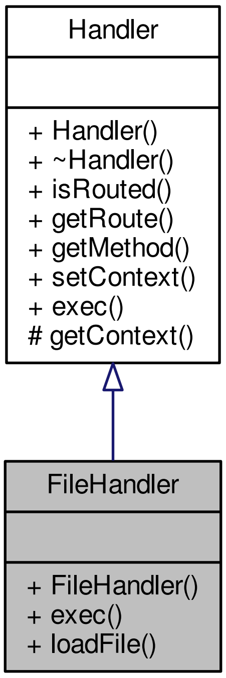
this class allow you to set any file of filesystem as response body

```
#include <file_handler.h>
```

Inheritance diagram for FileHandler:



Collaboration diagram for FileHandler:



Public Member Functions

- FileHandler (const char ∗route, const char ∗filePath, const char ∗mimeType, bool isBinary)
- void exec ()

Static Public Member Functions

- static bool loadFile (const char ∗filePath, std::string &data)

Additional Inherited Members

4.7.1 Detailed Description

this class allow you to set any file of filesystem as response body

FileHandler can handle as text files (like css, js), as binary data (img, png others)

4.7.2 Constructor & Destructor Documentation

4.7.2.1 FileHandler()

```
FileHandler::FileHandler (
            const char * route,
            const char * filePath,
            const char * mimeType,
            bool isBinary )
```

create file handlers with specified uri route, file path, content type etc.

Parameters

route	uri route file
filePath	local file path
mimeType	content type
isBinary	if file is binary, set true, false if it's text.

4.7.3 Member Function Documentation

4.7.3.1 exec()

```
void FileHandler::exec ( )  [virtual]
```

make http body of response object as file in filePath

Implements Handler.

4.7.3.2 loadFile()

```
bool FileHandler::loadFile (
            const char * filePath,
            std::string & data )  [static]
```

static function that read all data from file to string

Parameters

filePath	path to file
data	out param, if can read file, it will be written to data,do nothing otherwise

Returns

true, if read successfully, false otherwise

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

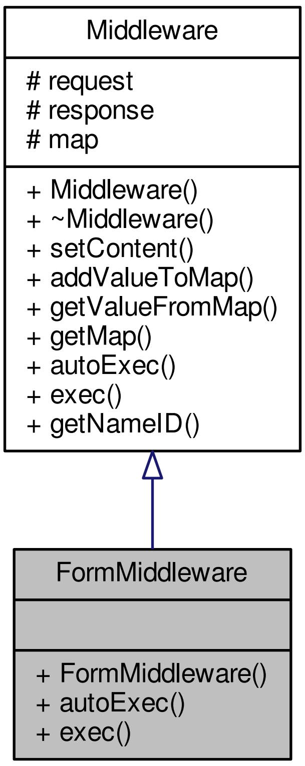
- /home/a_krava/projects/progbase3/web_server/include/file_handler.h
- /home/a_krava/projects/progbase3/web_server/src/file_handler.cpp

4.8 FormMiddleware Class Reference

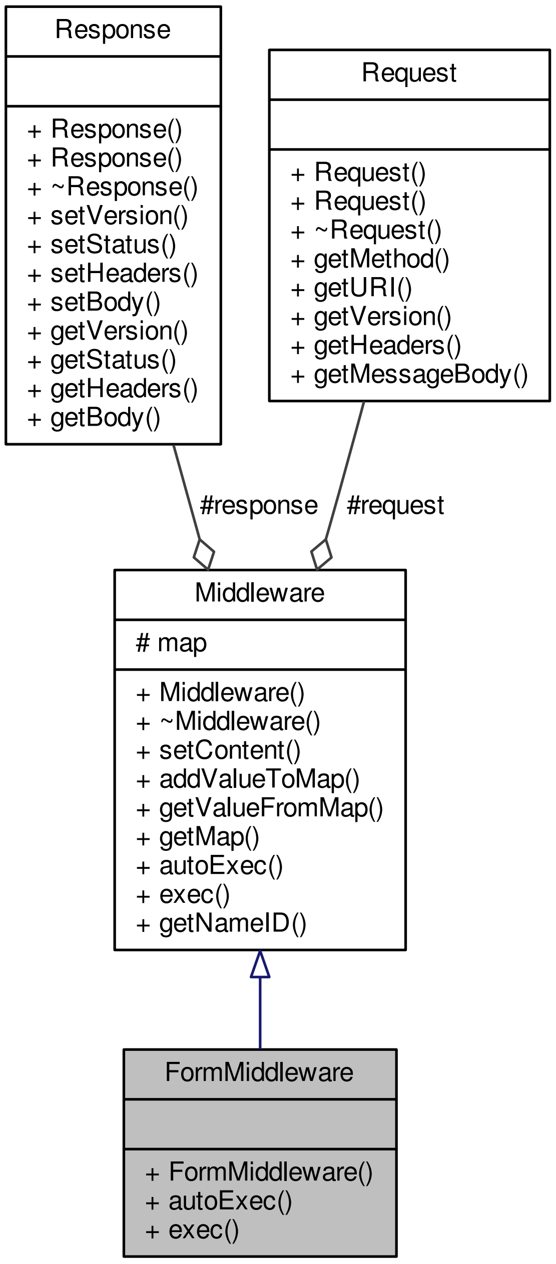
inherited class to parse application/x-www-form-urlencoded

```
#include <form_middleware.h>
```

Inheritance diagram for FormMiddleware:



Collaboration diagram for FormMiddleware:



Public Member Functions

- FormMiddleware (const char ∗nameID)
- bool autoExec ()
- void exec ()

Additional Inherited Members

4.8.1 Detailed Description

inherited class to parse application/x-www-form-urlencoded

FormMiddleware is intended to parse forms from http request and decode it

4.8.2 Constructor & Destructor Documentation

4.8.2.1 FormMiddleware()

```
FormMiddleware::FormMiddleware (
            const char * nameID )   [inline]
```

create middleware

Parameters

nameID	name id

4.8.3 Member Function Documentation

4.8.3.1 autoExec()

```
bool FormMiddleware::autoExec ( )   [virtual]
```

Check if request is application/x-www-form-urlencoded

Returns

true, if content type of http request is form

Implements Middleware.

4.8.3.2 exec()

```
void FormMiddleware::exec ( )   [virtual]
```

parse form in http request

Implements Middleware.

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

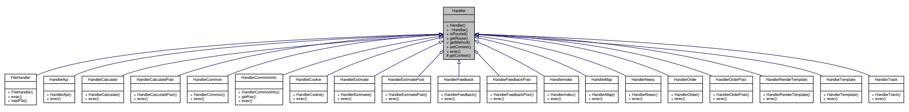
- /home/a_krava/projects/progbase3/web_server/include/form_middleware.h
- /home/a_krava/projects/progbase3/web_server/src/form_middleware.cpp

4.9 Handler Class Reference

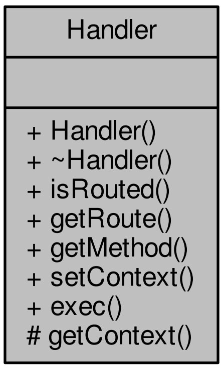
object of this class executes every time on new request, this object (and others) construct response to client

```
#include <handler.h>
```

Inheritance diagram for Handler:



Collaboration diagram for Handler:



Public Member Functions

- Handler (const char ∗route=nullptr, HTTP::Method method=HTTP::Method::ANY)
- virtual ∼Handler ()
- bool isRouted ()
- std::string getRoute ()
- HTTP::Method getMethod ()
- void setContext (Context ∗context)
- virtual void exec ()=0

Protected Member Functions

- Context ∗ getContext ()

4.9.1 Detailed Description

object of this class executes every time on new request, this object (and others) construct response to client

Handler object can be common (will execute on every response) or adjusted to some specified uri path. It can get all info about request, use added middleware, and make response

4.9.2 Constructor & Destructor Documentation

4.9.2.1 Handler()

```
Handler::Handler (
             const char * route = nullptr,
             HTTP::Method method = HTTP::Method::ANY )
```

create handler with params: common or routed one

Parameters

route	uri path route, if null - handler will be common
method	uri method, if ANY will be executed on any methods

4.9.2.2 ∼Handler()

```
virtual Handler::∼Handler ( )  [inline], [virtual]
```

destructs local variables

4.9.3 Member Function Documentation

4.9.3.1 exec()

```
virtual void Handler::exec ( )  [pure virtual]
```

this method will be executed on every request (or uri path if set)

Implemented in HandlerApi, HandlerFeedbackPost, HandlerFeedback, HandlerNews, HandlerCommonInfo, HandlerOrderPost, HandlerOrder, HandlerMap, HandlerEstimatePost, HandlerEstimate, HandlerCalculatePost, HandlerCalculate, HandlerTrack, HandlerIndex, HandlerRenderTemplate, HandlerCookie, HandlerTemplate, FileHandler, and HandlerCommon.

4.9.3.2 getContext()

```
Context * Handler::getContext ( )  [protected]
```

get current context

Returns

current Context object

4.9.3.3 getMethod()

```
HTTP::Method Handler::getMethod ( )
```

get Context object

Returns

current Context object

4.9.3.4 getRoute()

```
std::string Handler::getRoute ( )
```

get route of handler

Returns

uri http route path

4.9.3.5 isRouted()

```
bool Handler::isRouted ( )
```

check, if route is set

Returns

true, if handler for specified route, false if it's common one

4.9.3.6 setContext()

```
void Handler::setContext (
             Context * context )
```

set Context object

Parameters

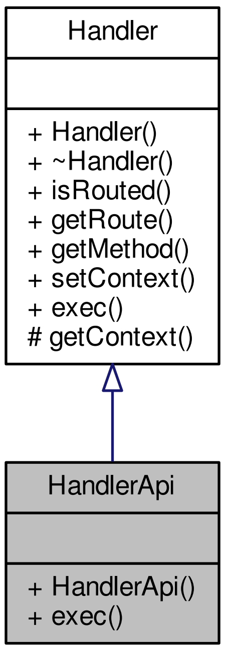
context	Context object

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

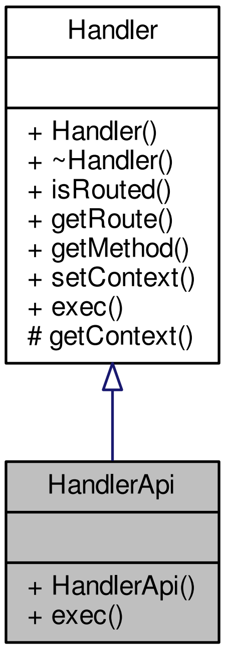
- /home/a_krava/projects/progbase3/web_server/include/handler.h
- /home/a_krava/projects/progbase3/web_server/src/handler.cpp

4.10 HandlerApi Class Reference

Inheritance diagram for HandlerApi:



Collaboration diagram for HandlerApi:



Public Member Functions

- HandlerApi (const char ∗ds, HTTP::Method m)
- void exec ()

Additional Inherited Members

4.10.1 Constructor & Destructor Documentation

4.10.1.1 HandlerApi()

```
HandlerApi::HandlerApi (
            const char ∗ ds,
            HTTP::Method m )  [inline]
```

4.10.2 Member Function Documentation

4.10.2.1 exec()

```
void HandlerApi::exec ( )  [inline], [virtual]
```

this method will be executed on every request (or uri path if set)

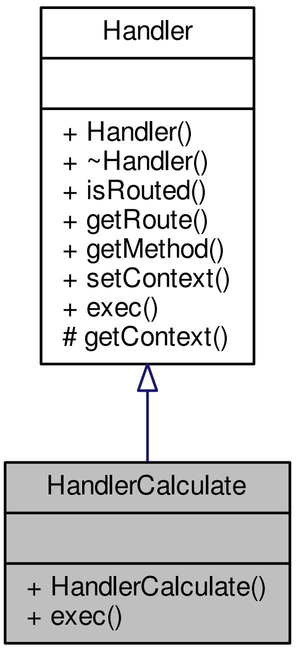
Implements Handler.

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

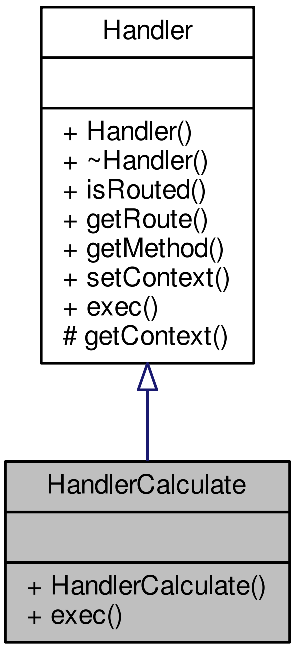
- /home/a_krava/projects/progbase3/web_server/main.cpp

4.11 HandlerCalculate Class Reference

Inheritance diagram for HandlerCalculate:



Collaboration diagram for HandlerCalculate:



Public Member Functions

- HandlerCalculate (const char ∗ds, HTTP::Method m)
- void exec ()

Additional Inherited Members

4.11.1 Constructor & Destructor Documentation

4.11.1.1 HandlerCalculate()

```
HandlerCalculate::HandlerCalculate (
            const char * ds,
            HTTP::Method m ) [inline]
```

4.11.2 Member Function Documentation

4.11.2.1 exec()

```
void HandlerCalculate::exec ( )  [inline], [virtual]
```

this method will be executed on every request (or uri path if set)

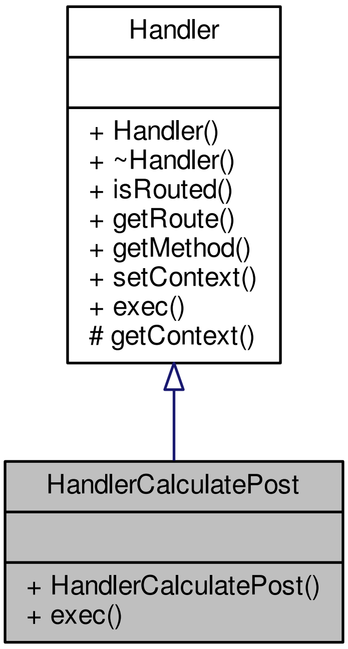
Implements Handler.

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

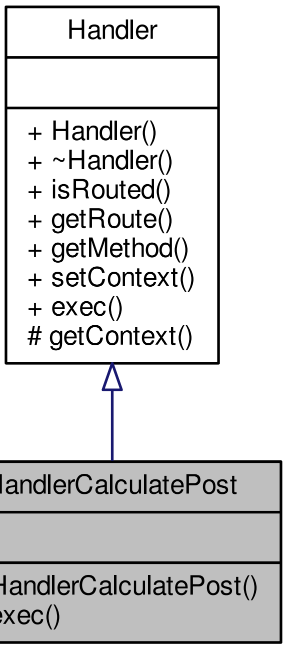
- /home/a_krava/projects/progbase3/web_server/main.cpp

4.12 HandlerCalculatePost Class Reference

Inheritance diagram for HandlerCalculatePost:



Collaboration diagram for HandlerCalculatePost:



Public Member Functions

- HandlerCalculatePost (const char ∗ds, HTTP::Method m)
- void exec ()

Additional Inherited Members

4.12.1 Constructor & Destructor Documentation

4.12.1.1 HandlerCalculatePost()

```
HandlerCalculatePost::HandlerCalculatePost (
            const char * ds,
            HTTP::Method m )  [inline]
```

4.12.2 Member Function Documentation

4.12.2.1 exec()

```
void HandlerCalculatePost::exec ( )  [inline], [virtual]
```

this method will be executed on every request (or uri path if set)

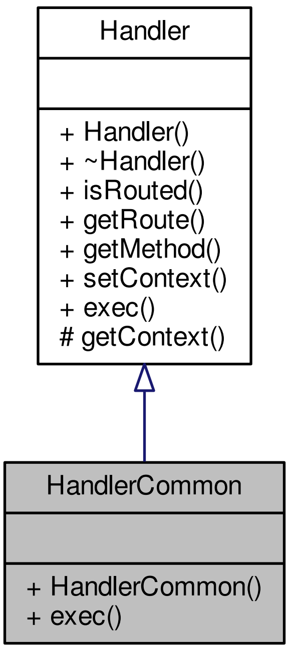
Implements Handler.

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

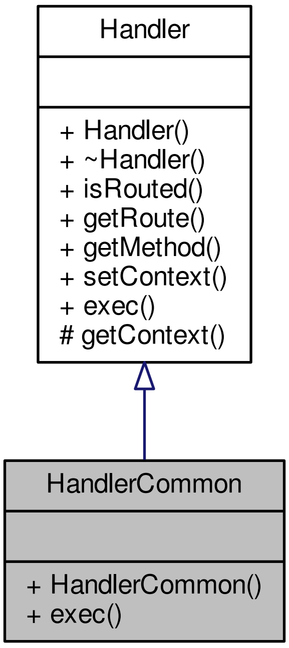
- /home/a_krava/projects/progbase3/web_server/main.cpp

4.13 HandlerCommon Class Reference

Inheritance diagram for HandlerCommon:



Collaboration diagram for HandlerCommon:



Public Member Functions

- HandlerCommon ()
- void exec ()

Additional Inherited Members

4.13.1 Constructor & Destructor Documentation

4.13.1.1 HandlerCommon()

```
HandlerCommon::HandlerCommon ( )  [inline]
```

4.13.2 Member Function Documentation

4.13.2.1 exec()

```
void HandlerCommon::exec ( )  [inline], [virtual]
```

this method will be executed on every request (or uri path if set)

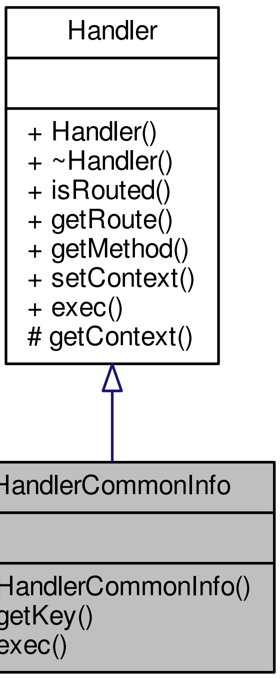
Implements Handler.

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

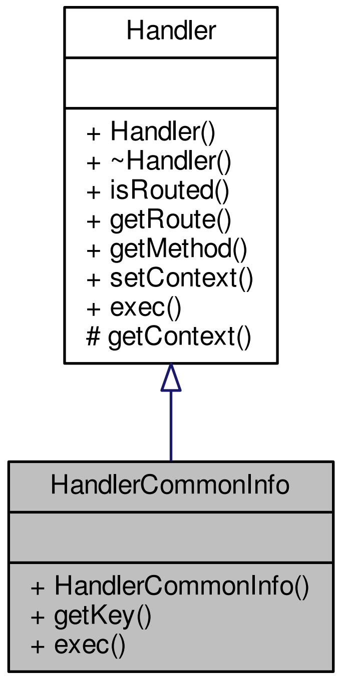
- /home/a_krava/projects/progbase3/web_server/main.cpp

4.14 HandlerCommonInfo Class Reference

Inheritance diagram for HandlerCommonInfo:



Collaboration diagram for HandlerCommonInfo:



Public Member Functions

- HandlerCommonInfo (const char ∗key_in_db, const char ∗ds, HTTP::Method m)
- const char ∗ getKey ()
- void exec ()

Additional Inherited Members

4.14.1 Constructor & Destructor Documentation

4.14.1.1 HandlerCommonInfo()

```
HandlerCommonInfo::HandlerCommonInfo (
             const char * key_in_db,
             const char * ds,
             HTTP::Method m )  [inline]
```

4.14.2 Member Function Documentation

4.14.2.1 exec()

```
void HandlerCommonInfo::exec ( )  [inline], [virtual]
```

this method will be executed on every request (or uri path if set)

Implements Handler.

4.14.2.2 getKey()

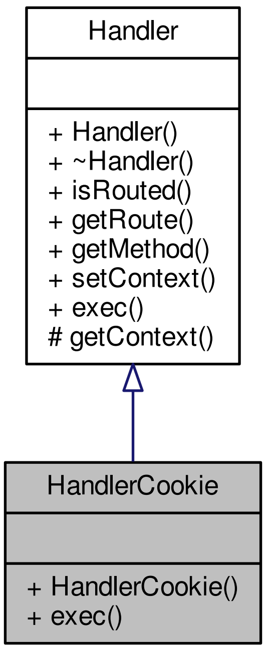
```
const char* HandlerCommonInfo::getKey ( )  [inline]
```

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

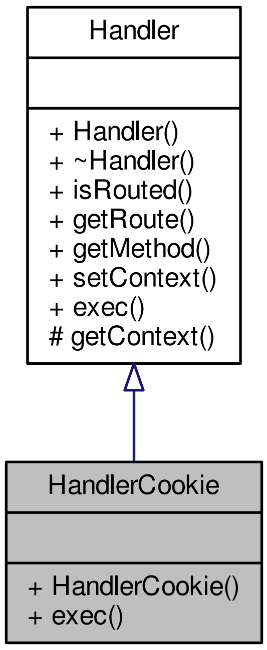
- /home/a_krava/projects/progbase3/web_server/main.cpp

4.15 HandlerCookie Class Reference

Inheritance diagram for HandlerCookie:



Collaboration diagram for HandlerCookie:



Public Member Functions

- HandlerCookie ()
- void exec ()

Additional Inherited Members

4.15.1 Constructor & Destructor Documentation

4.15.1.1 HandlerCookie()

```
HandlerCookie::HandlerCookie ( )  [inline]
```

4.15.2 Member Function Documentation

4.15.2.1 exec()

```
void HandlerCookie::exec ( )  [inline], [virtual]
```

this method will be executed on every request (or uri path if set)

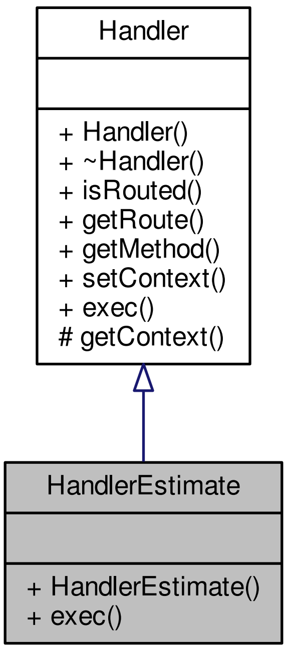
Implements Handler.

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

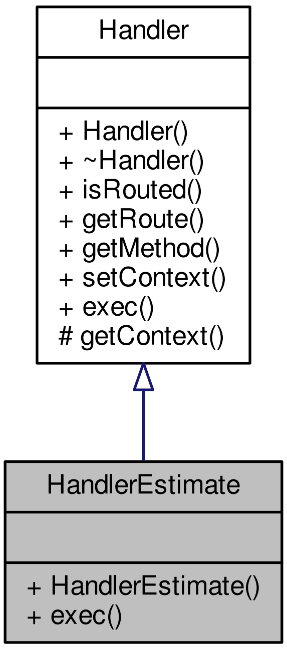
- /home/a_krava/projects/progbase3/web_server/main.cpp

4.16 HandlerEstimate Class Reference

Inheritance diagram for HandlerEstimate:



Collaboration diagram for HandlerEstimate:



Public Member Functions

- HandlerEstimate (const char ∗ds, HTTP::Method m)
- void exec ()

Additional Inherited Members

4.16.1 Constructor & Destructor Documentation

4.16.1.1 HandlerEstimate()

```
HandlerEstimate::HandlerEstimate (
            const char * ds,
            HTTP::Method m )  [inline]
```

4.16.2 Member Function Documentation

4.16.2.1 exec()

```
void HandlerEstimate::exec ( )  [inline], [virtual]
```

this method will be executed on every request (or uri path if set)

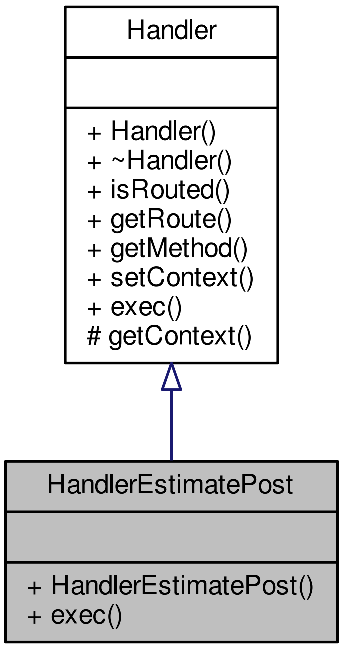
Implements Handler.

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

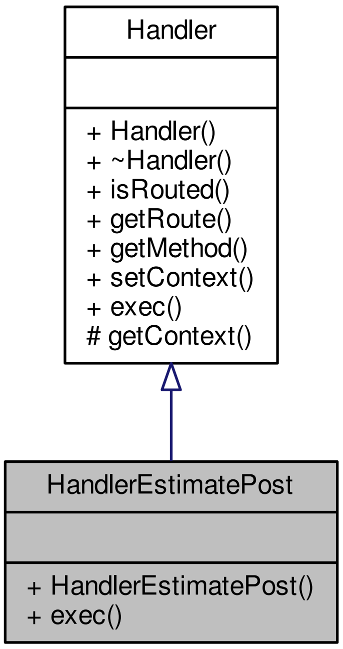
- /home/a_krava/projects/progbase3/web_server/main.cpp

4.17 HandlerEstimatePost Class Reference

Inheritance diagram for HandlerEstimatePost:



Collaboration diagram for HandlerEstimatePost:



Public Member Functions

- HandlerEstimatePost (const char *ds, HTTP::Method m)
- void exec ()

Additional Inherited Members

4.17.1 Constructor & Destructor Documentation

4.17.1.1 HandlerEstimatePost()

```
HandlerEstimatePost::HandlerEstimatePost (
            const char * ds,
            HTTP::Method m )  [inline]
```

4.17.2 Member Function Documentation

4.17.2.1 exec()

```
void HandlerEstimatePost::exec ( )  [inline], [virtual]
```

this method will be executed on every request (or uri path if set)

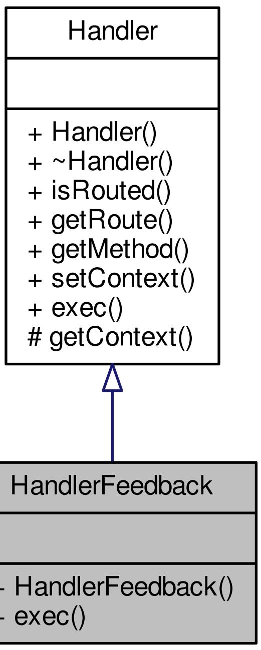
Implements Handler.

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

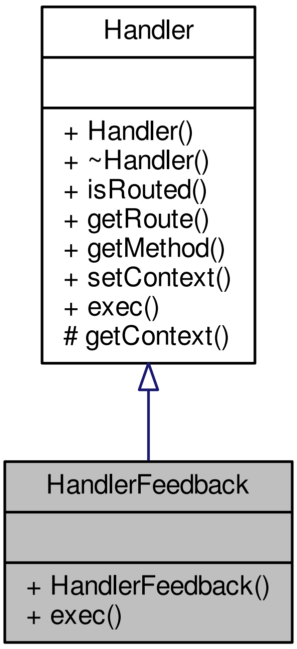
- /home/a_krava/projects/progbase3/web_server/main.cpp

4.18 HandlerFeedback Class Reference

Inheritance diagram for HandlerFeedback:



Collaboration diagram for HandlerFeedback:



Public Member Functions

- HandlerFeedback (const char ∗ds, HTTP::Method m)
- void exec ()

Additional Inherited Members

4.18.1 Constructor & Destructor Documentation

4.18.1.1 HandlerFeedback()

```
HandlerFeedback::HandlerFeedback (
            const char ∗ ds,
            HTTP::Method m )  [inline]
```

4.18.2 Member Function Documentation

4.18.2.1 exec()

```
void HandlerFeedback::exec ( )  [inline], [virtual]
```

this method will be executed on every request (or uri path if set)

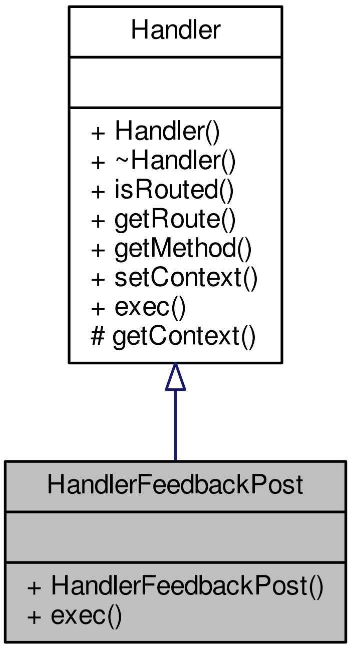
Implements Handler.

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

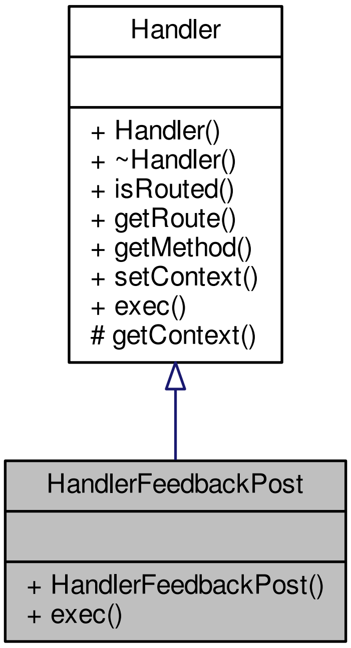
- /home/a_krava/projects/progbase3/web_server/main.cpp

4.19 HandlerFeedbackPost Class Reference

Inheritance diagram for HandlerFeedbackPost:



Collaboration diagram for HandlerFeedbackPost:



Public Member Functions

- HandlerFeedbackPost (const char ∗ds, HTTP::Method m)
- void exec ()

Additional Inherited Members

4.19.1 Constructor & Destructor Documentation

4.19.1.1 HandlerFeedbackPost()

```
HandlerFeedbackPost::HandlerFeedbackPost (
            const char ∗ ds,
            HTTP::Method m )  [inline]
```

4.19.2 Member Function Documentation

4.19.2.1 exec()

```
void HandlerFeedbackPost::exec ( )  [inline], [virtual]
```

this method will be executed on every request (or uri path if set)

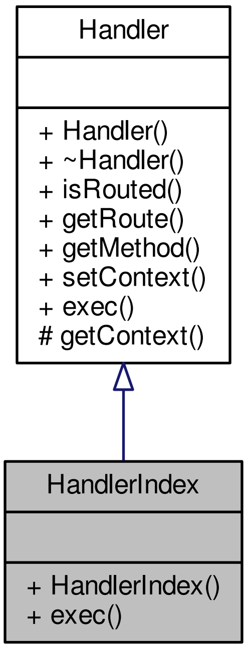
Implements Handler.

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

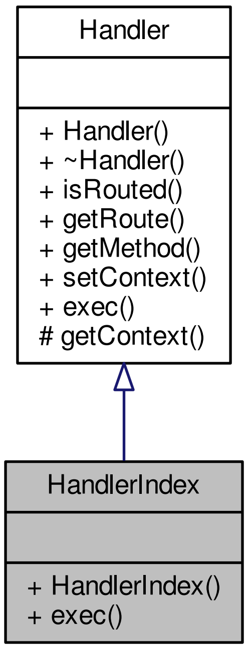
- /home/a_krava/projects/progbase3/web_server/main.cpp

4.20 HandlerIndex Class Reference

Inheritance diagram for HandlerIndex:



Collaboration diagram for HandlerIndex:



Public Member Functions

- HandlerIndex (const char ∗ds, HTTP::Method m)
- void exec ()

Additional Inherited Members

4.20.1 Constructor & Destructor Documentation

4.20.1.1 HandlerIndex()

```
HandlerIndex::HandlerIndex (
            const char ∗ ds,
            HTTP::Method m )  [inline]
```

4.20.2 Member Function Documentation

4.20.2.1 exec()

```
void HandlerIndex::exec ( )  [inline], [virtual]
```

this method will be executed on every request (or uri path if set)

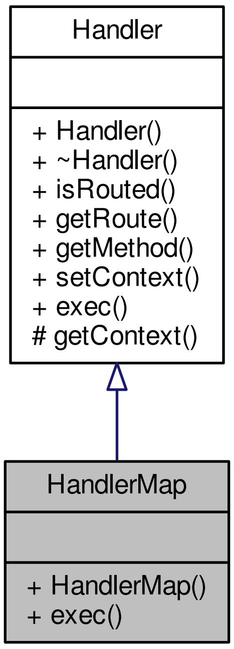
Implements Handler.

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

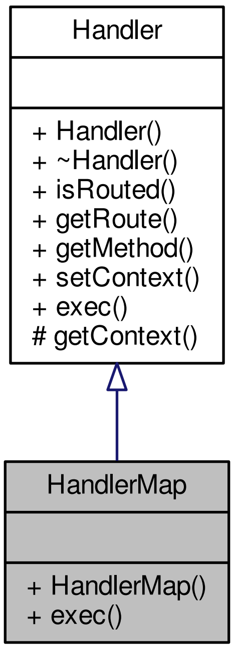
- /home/a_krava/projects/progbase3/web_server/main.cpp

4.21 HandlerMap Class Reference

Inheritance diagram for HandlerMap:



Collaboration diagram for HandlerMap:



Public Member Functions

- HandlerMap (const char ∗ds, HTTP::Method m)
- void exec ()

Additional Inherited Members

4.21.1 Constructor & Destructor Documentation

4.21.1.1 HandlerMap()

```
HandlerMap::HandlerMap (
             const char * ds,
             HTTP::Method m )  [inline]
```

4.21.2 Member Function Documentation

4.21.2.1 exec()

```
void HandlerMap::exec ( )  [inline], [virtual]
```

this method will be executed on every request (or uri path if set)

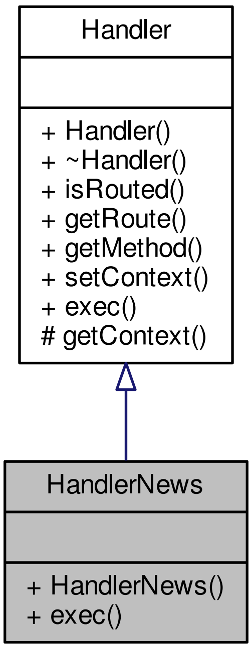
Implements Handler.

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

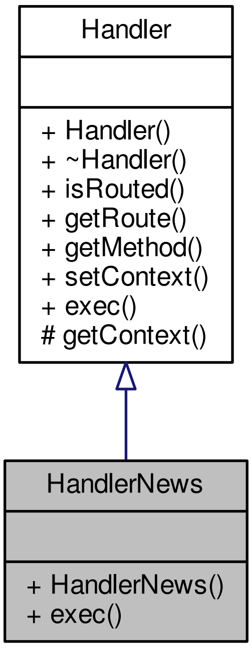
- /home/a_krava/projects/progbase3/web_server/main.cpp

4.22 HandlerNews Class Reference

Inheritance diagram for HandlerNews:



Collaboration diagram for HandlerNews:



Public Member Functions

- HandlerNews (const char ∗ds, HTTP::Method m)
- void exec ()

Additional Inherited Members

4.22.1 Constructor & Destructor Documentation

4.22.1.1 HandlerNews()

```
HandlerNews::HandlerNews (
             const char ∗ ds,
             HTTP::Method m )  [inline]
```

4.22.2 Member Function Documentation

4.22.2.1 exec()

```
void HandlerNews::exec ( )  [inline], [virtual]
```

this method will be executed on every request (or uri path if set)

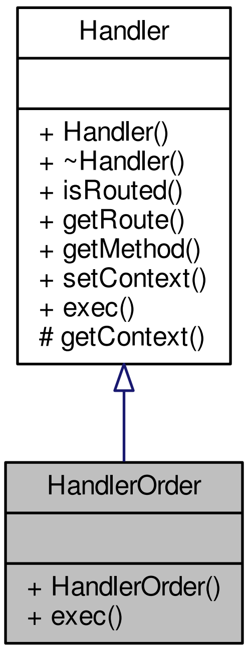
Implements Handler.

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

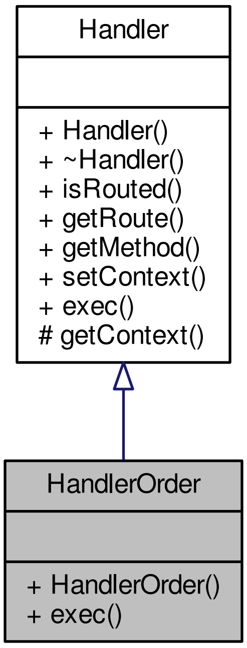
- /home/a_krava/projects/progbase3/web_server/main.cpp

4.23 HandlerOrder Class Reference

Inheritance diagram for HandlerOrder:



Collaboration diagram for HandlerOrder:



Public Member Functions

- HandlerOrder (const char ∗ds, HTTP::Method m)
- void exec ()

Additional Inherited Members

4.23.1 Constructor & Destructor Documentation

4.23.1.1 HandlerOrder()

```
HandlerOrder::HandlerOrder (
             const char * ds,
             HTTP::Method m )  [inline]
```

4.23.2 Member Function Documentation

4.23.2.1 exec()

```
void HandlerOrder::exec ( )  [inline], [virtual]
```

this method will be executed on every request (or uri path if set)

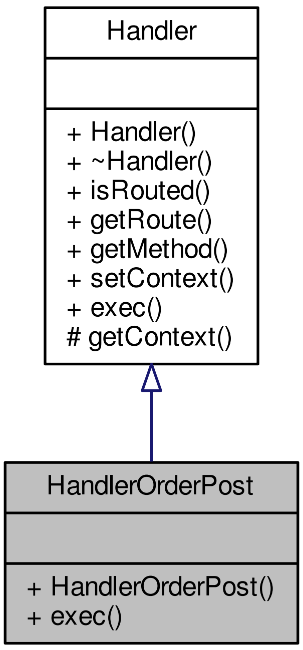
Implements Handler.

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

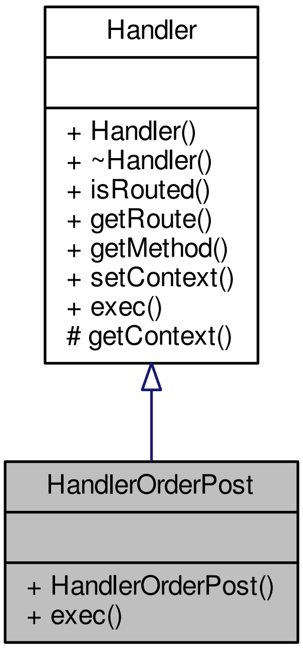
- /home/a_krava/projects/progbase3/web_server/main.cpp

4.24 HandlerOrderPost Class Reference

Inheritance diagram for HandlerOrderPost:



Collaboration diagram for HandlerOrderPost:



Public Member Functions

- HandlerOrderPost (const char ∗ds, HTTP::Method m)
- void exec ()

Additional Inherited Members

4.24.1 Constructor & Destructor Documentation

4.24.1.1 HandlerOrderPost()

```
HandlerOrderPost::HandlerOrderPost (
            const char * ds,
            HTTP::Method m )  [inline]
```

4.24.2 Member Function Documentation

4.24.2.1 exec()

```
void HandlerOrderPost::exec ( )  [inline], [virtual]
```

this method will be executed on every request (or uri path if set)

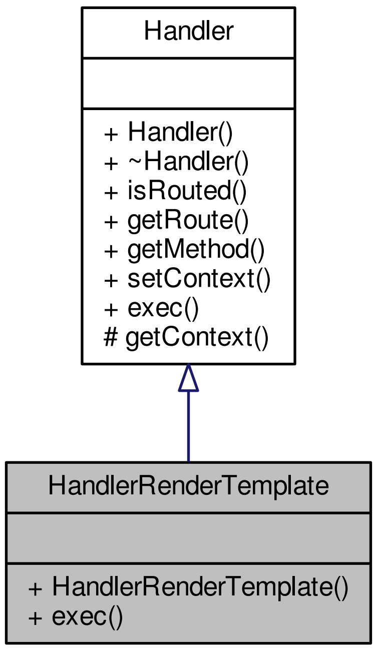
Implements Handler.

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

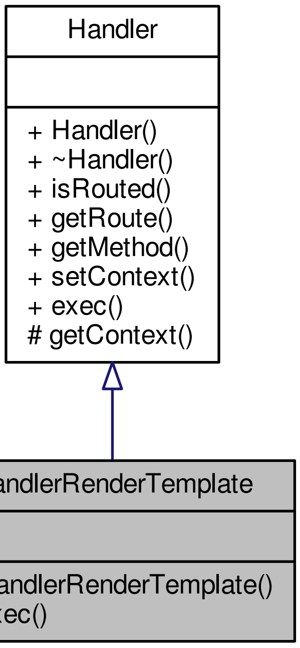
- /home/a_krava/projects/progbase3/web_server/main.cpp

4.25 HandlerRenderTemplate Class Reference

Inheritance diagram for HandlerRenderTemplate:



Collaboration diagram for HandlerRenderTemplate:



Public Member Functions

- HandlerRenderTemplate ()
- void exec ()

Additional Inherited Members

4.25.1 Constructor & Destructor Documentation

4.25.1.1 HandlerRenderTemplate()

```
HandlerRenderTemplate::HandlerRenderTemplate ( )  [inline]
```

4.25.2 Member Function Documentation

4.25.2.1 exec()

```
void HandlerRenderTemplate::exec ( )  [inline], [virtual]
```

this method will be executed on every request (or uri path if set)

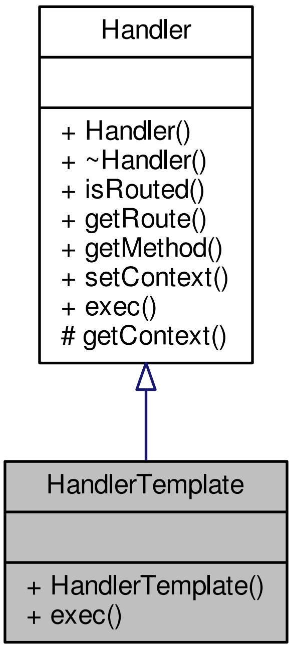
Implements Handler.

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

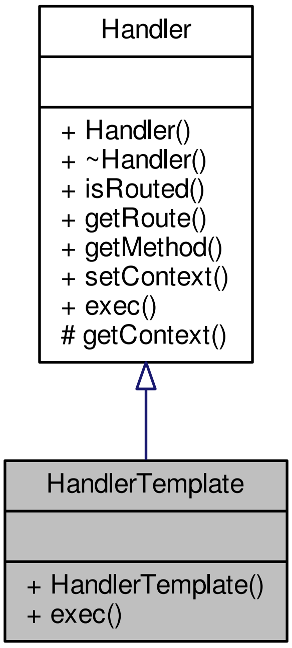
- /home/a_krava/projects/progbase3/web_server/main.cpp

4.26 HandlerTemplate Class Reference

Inheritance diagram for HandlerTemplate:



Collaboration diagram for HandlerTemplate:



Public Member Functions

- HandlerTemplate ()
- void exec ()

Additional Inherited Members

4.26.1 Constructor & Destructor Documentation

4.26.1.1 HandlerTemplate()

```
HandlerTemplate::HandlerTemplate ( )  [inline]
```

4.26.2 Member Function Documentation

4.26.2.1 exec()

```
void HandlerTemplate::exec ( )  [inline], [virtual]
```

this method will be executed on every request (or uri path if set)

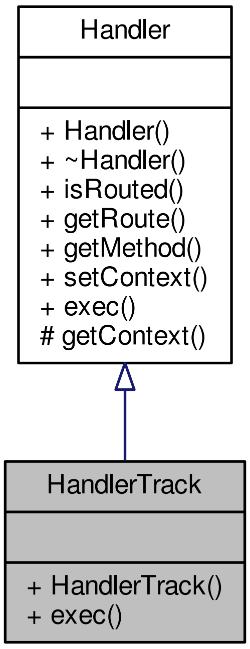
Implements Handler.

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

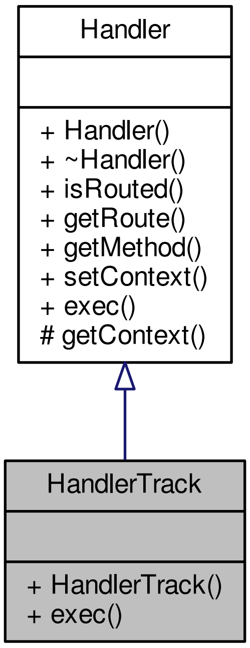
- /home/a_krava/projects/progbase3/web_server/main.cpp

4.27 HandlerTrack Class Reference

Inheritance diagram for HandlerTrack:



Collaboration diagram for HandlerTrack:



Public Member Functions

- HandlerTrack (const char ∗ds, HTTP::Method m)
- void exec ()

Additional Inherited Members

4.27.1 Constructor & Destructor Documentation

4.27.1.1 HandlerTrack()

```
HandlerTrack::HandlerTrack (
            const char * ds,
            HTTP::Method m )  [inline]
```

4.27.2 Member Function Documentation

4.27.2.1 exec()

```
void HandlerTrack::exec ( )  [inline], [virtual]
```

this method will be executed on every request (or uri path if set)

Implements Handler.

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

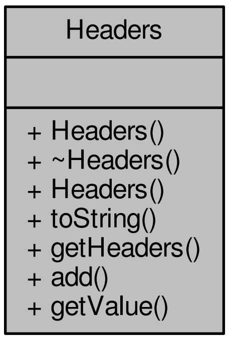
- /home/a_krava/projects/progbase3/web_server/main.cpp

4.28 Headers Class Reference

wrapper class for http headers

```
#include <headers.h>
```

Collaboration diagram for Headers:



Public Member Functions

- Headers ()
- ∼Headers ()
- Headers (std::string &httpHeaders)
- std::string toString ()
- std::unordered_map< std::string, std::string > getHeaders ()
- void add (const char ∗key, const char ∗value)
- bool getValue (const char ∗key, std::string &value)

4.28.1 Detailed Description

wrapper class for http headers

Headers consist of map with key-value pairs, and is using for Request and Response http objects

4.28.2 Constructor & Destructor Documentation

4.28.2.1 Headers() [1/2]

```
Headers::Headers ( )
```

create empty headers object

4.28.2.2 ∼Headers()

```
Headers::∼Headers ( )
```

cleanup map of key-value pairs

4.28.2.3 Headers() [2/2]

```
Headers::Headers (
            std::string & httpHeaders )
```

create Headers, parsing http input string

Parameters

httpHeaders	input http headers string

4.28.3 Member Function Documentation

4.28.3.1 add()

```
void Headers::add (
            const char * key,
            const char * value )
```

insert value by key to map, if key exists, it will be overwritten

Parameters

key	input key
value	input value

4.28.3.2 getHeaders()

```
unordered_map< string, string > Headers::getHeaders ( )
```

get current map

Returns

map of key-value pairs

4.28.3.3 getValue()

```
bool Headers::getValue (
            const char * key,
            std::string & value )
```

get value from map by key

Parameters

key	searched key
value	out param, if key exists, value will be written, nothing do otherwise

Returns

true if value exists, false otherwise

4.28.3.4 toString()

```
std::string Headers::toString ( )
```

serialize Headers to string

Returns

serialized string

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

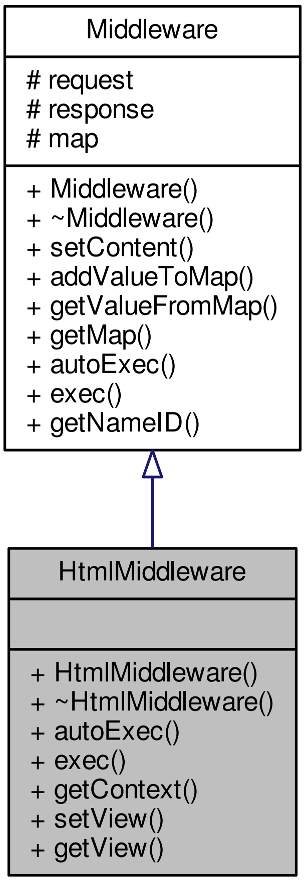
- /home/a_krava/projects/progbase3/web_server/include/headers.h
- /home/a_krava/projects/progbase3/web_server/src/headers.cpp

4.29 HtmlMiddleware Class Reference

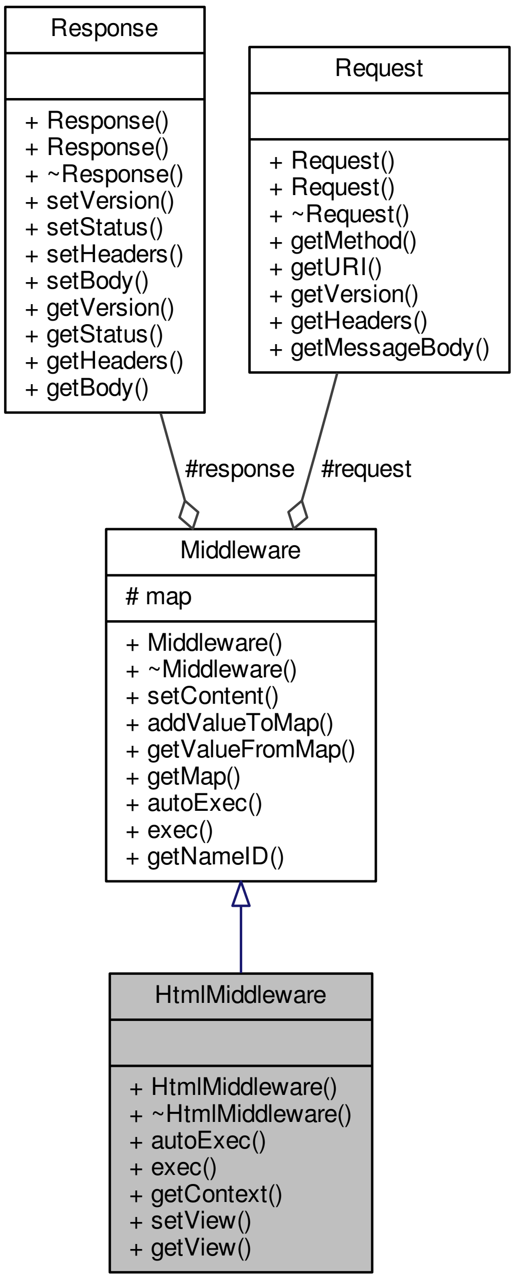
inherited class to render html pages from templates

```
#include <html_middleware.h>
```

Inheritance diagram for HtmlMiddleware:



Collaboration diagram for HtmlMiddleware:



Public Member Functions

- HtmlMiddleware (const char ∗nameID)
- ∼HtmlMiddleware ()
- bool autoExec ()
- void exec ()
- mstch::map ∗ getContext ()

- void setView (std::string &view)
- std::string getView ()

Additional Inherited Members

4.29.1 Detailed Description

inherited class to render html pages from templates

HtmlMiddleware uses logic-less mustache templates to render html pages

4.29.2 Constructor & Destructor Documentation

4.29.2.1 HtmlMiddleware()

```
HtmlMiddleware::HtmlMiddleware (
             const char * nameID )
```

create middleware

Parameters

nameID	name id

4.29.2.2 ∼HtmlMiddleware()

```
HtmlMiddleware::∼HtmlMiddleware ( )
```

delete context map, used fot rendering

4.29.3 Member Function Documentation

4.29.3.1 autoExec()

```
bool HtmlMiddleware::autoExec ( )  [virtual]
```

Cleanup context map

Returns

true, if ready to render

Implements Middleware.

4.29.3.2 exec()

```
void HtmlMiddleware::exec ( )  [virtual]
```

render template and set to response body

Implements Middleware.

4.29.3.3 getContext()

```
mstch::map * HtmlMiddleware::getContext ( )
```

get current context map

Returns

context map of template

4.29.3.4 getView()

```
std::string HtmlMiddleware::getView ( )
```

get current template view

Returns

template view string

4.29.3.5 setView()

```
void HtmlMiddleware::setView (
             std::string & view )
```

set new template view

Parameters

view	template view as string

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

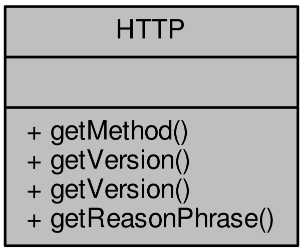
- /home/a_krava/projects/progbase3/web_server/include/html_middleware.h
- /home/a_krava/projects/progbase3/web_server/src/html_middleware.cpp

4.30 HTTP Class Reference

static class describes http method, version, and allow to convert it from/to string/enumeration

```
#include <http.h>
```

Collaboration diagram for HTTP:



Public Types

- enum Method {
  UNDEFINED, GET, HEAD, POST,
  PUT, DELETE, CONNECT, OPTIONS,
  TRACE, PATCH, ANY }
- enum Version {
  HTTP_UNDEFINED, HTTP_0_9, HTTP_1_0, HTTP_1_1,
  HTTP_2_0, HTTP_ANY }

Static Public Member Functions

- static HTTP::Method getMethod (std::string &str)
- static HTTP::Version getVersion (std::string &str)
- static std::string getVersion (HTTP::Version version)
- static std::string getReasonPhrase (int code)

4.30.1 Detailed Description

static class describes http method, version, and allow to convert it from/to string/enumeration

HTTP class describes Method, Version, ReasonPhrase of code in http

4.30.2 Member Enumeration Documentation

4.30.2.1 Method

```
enum HTTP::Method
```

Flags to define combinations of HTTP Request methods

Enumerator

UNDEFINED	
GET	
HEAD	
POST	
PUT	
DELETE	
CONNECT	
OPTIONS	
TRACE	
PATCH	
ANY	

4.30.2.2 Version

```
enum HTTP::Version
```

Flags to define combinations of HTTP Version

Enumerator

HTTP_UNDEFINED	
HTTP_0_9	
HTTP_1_0	
HTTP_1_1	
HTTP_2_0	
HTTP_ANY	

4.30.3 Member Function Documentation

4.30.3.1 getMethod()

```
HTTP::Method HTTP::getMethod (
             std::string & str )  [static]
```

Parse input string to http method

Parameters

str	input string

Returns

parsed method from string, if string wasn't valid returns UNDEFINED

4.30.3.2 getReasonPhrase()

```
std::string HTTP::getReasonPhrase (
            int code )  [static]
```

Serialize status code to string

Parameters

code	http status code

Returns

reason phrase for code as string, returns Not Found if code not found among values

4.30.3.3 getVersion() [1/2]

```
HTTP::Version HTTP::getVersion (
            std::string & str )  [static]
```

Parse input string to http version

Parameters

str	input string

Returns

parsed version from string, if string wasn't valid returns HTTP_UNDEFINED

4.30.3.4 getVersion() [2/2]

```
std::string HTTP::getVersion (
            HTTP::Version version )  [static]
```

Serialize HTTP::Version to string

Parameters

version	http version

Returns

version as string

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

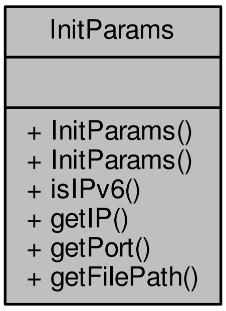
- /home/a_krava/projects/progbase3/web_server/include/http.h
- /home/a_krava/projects/progbase3/web_server/src/http.cpp

4.31 InitParams Class Reference

InitParams is intended to get web-server configs from command line arguments.

`#include <init_params.h>`

Collaboration diagram for InitParams:



Public Member Functions

- InitParams ()
- InitParams (int argc, char ∗∗argv)
- bool isIPv6 ()
- const char ∗ getIP ()
- int getPort ()
- std::string getFilePath ()

4.31.1 Detailed Description

InitParams is intended to get web-server configs from command line arguments.

This class make verification of ip-address, port etc...

4.31.2 Constructor & Destructor Documentation

4.31.2.1 InitParams() [1/2]

```
InitParams::InitParams ( )
```

Create empty object

4.31.2.2 InitParams() [2/2]

```
InitParams::InitParams (
             int argc,
             char ** argv )
```

Get params from command line arguments

Parameters

argc	num of params
argv	params

4.31.3 Member Function Documentation

4.31.3.1 getFilePath()

```
std::string InitParams::getFilePath ( )
```

get log file path

Returns

log file path

4.31.3.2 getIP()

```
const char * InitParams::getIP ( )
```

get ip address

Returns

ip host address

4.31.3.3 getPort()

```
int InitParams::getPort ( )
```

get port

Returns

host post

4.31.3.4 isIPv6()

```
bool InitParams::isIPv6 ( )
```

check if ip address is IPv6

Returns

true if ip address is IPv6, false otherwise

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

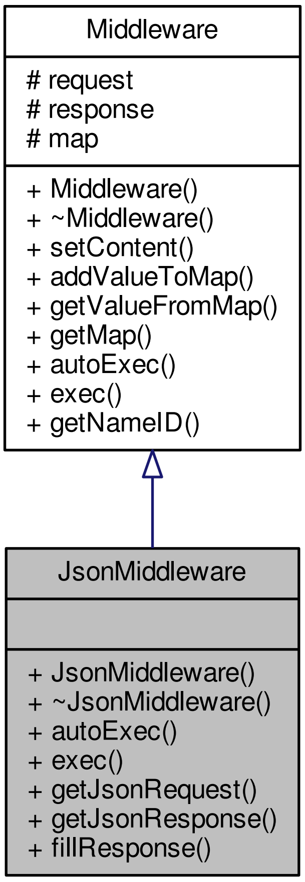
- /home/a_krava/projects/progbase3/web_server/include/init_params.h
- /home/a_krava/projects/progbase3/web_server/src/init_params.cpp

4.32 JsonMiddleware Class Reference

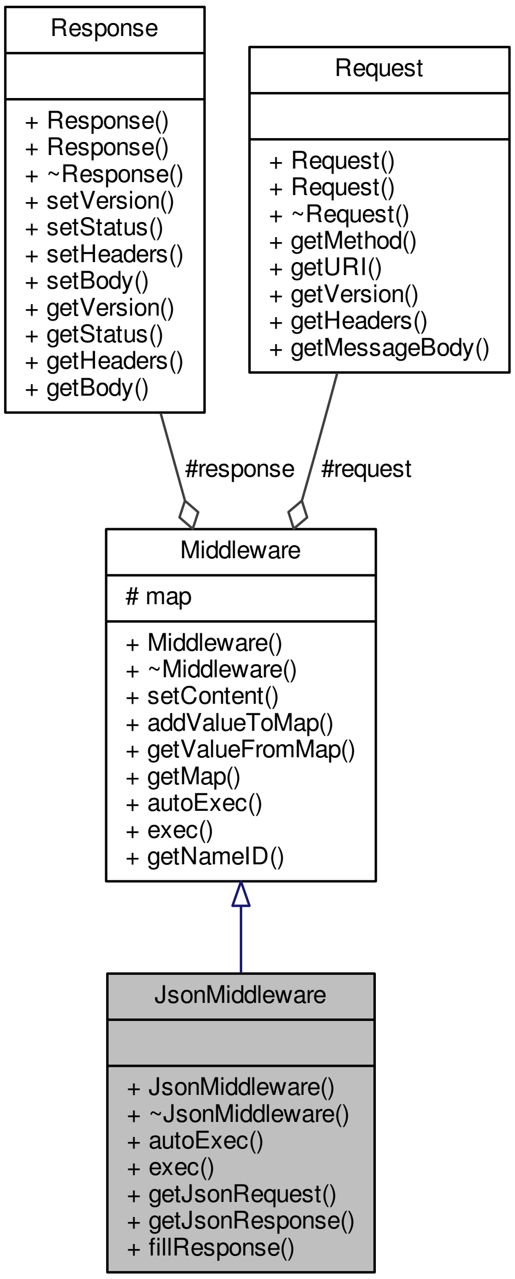
inherited class to perform any actions with json data

```
#include <json_middleware.h>
```

Inheritance diagram for JsonMiddleware:



Collaboration diagram for JsonMiddleware:



Public Member Functions

- JsonMiddleware (const char ∗nameID)
- ∼JsonMiddleware ()
- bool autoExec ()
- void exec ()
- nlohmann::json ∗ getJsonRequest ()

- nlohmann::json ∗ getJsonResponse ()
- void fillResponse ()

Additional Inherited Members

4.32.1 Detailed Description

inherited class to perform any actions with json data

JsonMiddleware is intended to parse json from http request, fill response with json and perform any actions with json

4.32.2 Constructor & Destructor Documentation

4.32.2.1 JsonMiddleware()

```
JsonMiddleware::JsonMiddleware (
            const char * nameID )
```

create middleware

Parameters

nameID	name id

4.32.2.2 ∼JsonMiddleware()

```
JsonMiddleware::∼JsonMiddleware ( )
```

delete json request and response objects

4.32.3 Member Function Documentation

4.32.3.1 autoExec()

```
bool JsonMiddleware::autoExec ( )  [virtual]
```

Check if request is json data

Returns

true, if content type of http request is json

Implements Middleware.

4.32.3.2 exec()

```
void JsonMiddleware::exec ( )  [virtual]
```

parse json from http request

Implements Middleware.

4.32.3.3 fillResponse()

```
void JsonMiddleware::fillResponse ( )
```

set response body with serialized json data from jsonResponse

4.32.3.4 getJsonRequest()

```
nlohmann::json * JsonMiddleware::getJsonRequest ( )
```

get json request object

Returns

json request object

4.32.3.5 getJsonResponse()

```
nlohmann::json * JsonMiddleware::getJsonResponse ( )
```

get json response object

Returns

json response object

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

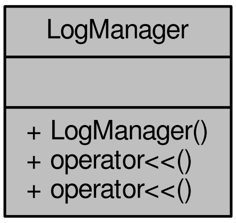
- /home/a_krava/projects/progbase3/web_server/include/json_middleware.h
- /home/a_krava/projects/progbase3/web_server/src/json_middleware.cpp

4.33 LogManager Class Reference

logging info into file

```
#include <log_manager.h>
```

Collaboration diagram for LogManager:



Public Member Functions

- LogManager (const char ∗fileName)
- void operator<< (const char ∗data)
- void operator<< (std::string data)

4.33.1 Detailed Description

logging info into file

LogManager create file and append it with input data data

4.33.2 Constructor & Destructor Documentation

4.33.2.1 LogManager()

```
LogManager::LogManager (
            const char * fileName )
```

create log file, if fileName is null no data will be written

Parameters

fileName	path to file (could be null)

4.33.3 Member Function Documentation

4.33.3.1 operator<<() [1/2]

```
void LogManager::operator<< (
             const char * data )
```

append to log new info

Parameters

data	logging information

4.33.3.2 operator<<() [2/2]

```
void LogManager::operator<< (
             std::string data )
```

append to log new info

Parameters

data	logging information

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

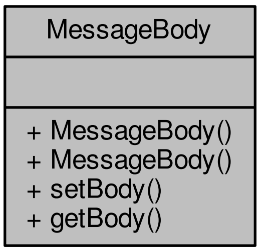
- /home/a_krava/projects/progbase3/web_server/include/log_manager.h
- /home/a_krava/projects/progbase3/web_server/src/log_manager.cpp

4.34 MessageBody Class Reference

wrapper class for http body

```
#include <message_body.h>
```

Collaboration diagram for MessageBody:



Public Member Functions

- MessageBody ()
- MessageBody (std::string &body)
- void setBody (std::string &body)
- std::string getBody ()

4.34.1 Detailed Description

wrapper class for http body

MessageBody contains decoded information about http body

4.34.2 Constructor & Destructor Documentation

4.34.2.1 MessageBody() [1/2]

```
MessageBody::MessageBody ( )
```

Create empty http body

4.34.2.2 MessageBody() [2/2]

```
MessageBody::MessageBody (
            std::string & body )
```

Create http body from input string

Parameters

body	input string

4.34.3 Member Function Documentation

4.34.3.1 getBody()

```
std::string MessageBody::getBody ( )
```

get http body as string

Returns

http body

4.34.3.2 setBody()

```
void MessageBody::setBody (
            std::string & body )
```

set http body as string

Parameters

body	http body

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

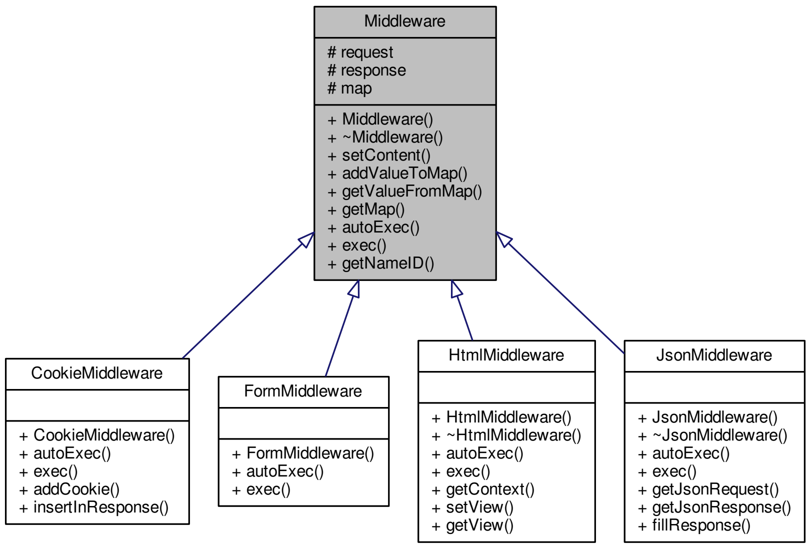
- /home/a_krava/projects/progbase3/web_server/include/message_body.h
- /home/a_krava/projects/progbase3/web_server/src/message_body.cpp

4.35 Middleware Class Reference

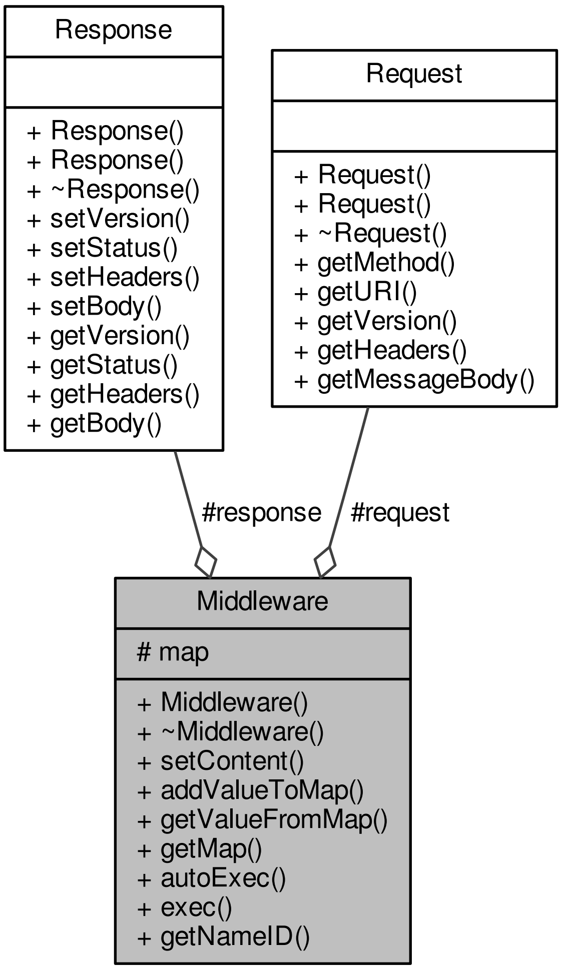
class wrapper for middleware

```
#include <middleware.h>
```

Inheritance diagram for Middleware:



Collaboration diagram for Middleware:



Public Member Functions

- Middleware (const char ∗nameID)
- virtual ∼Middleware ()
- void setContent (Request ∗request, Response ∗response)
- void addValueToMap (const char ∗key, const char ∗value)
- bool getValueFromMap (const char ∗key, std::string &value)
- std::unordered_map< std::string, std::string > ∗ getMap ()
- virtual bool autoExec ()=0
- virtual void exec ()=0
- std::string getNameID ()

Protected Attributes

- Request ∗ request
- Response ∗ response
- std::unordered_map< std::string, std::string > ∗ map

4.35.1 Detailed Description

class wrapper for middleware

Middleware have got current request and response objects and also map for key-value pairs. Method exec and autoExec can use request and response objects to perform actions.

4.35.2 Constructor & Destructor Documentation

4.35.2.1 Middleware()

```
Middleware::Middleware (
             const char * nameID )
```

Create empty middleware, where Request and Response objects are null

Parameters

nameID	name id as string

4.35.2.2 ∼Middleware()

```
Middleware::~Middleware ( )  [virtual]
```

4.35.3 Member Function Documentation

4.35.3.1 addValueToMap()

```
void Middleware::addValueToMap (
             const char * key,
             const char * value )
```

Add value to map by key. If key exists, it should be overwritten

Parameters

key	key as string
value	value as string

4.35.3.2 autoExec()

```
virtual bool Middleware::autoExec ( )  [pure virtual]
```

Check if current request allow do exec method

Returns

true, if need do exec with current request, false otherwise

Implemented in JsonMiddleware, HtmlMiddleware, CookieMiddleware, and FormMiddleware.

4.35.3.3 exec()

```
virtual void Middleware::exec ( )  [pure virtual]
```

perform operation with request and response objects

Implemented in JsonMiddleware, HtmlMiddleware, CookieMiddleware, and FormMiddleware.

4.35.3.4 getMap()

```
unordered_map< string, string > * Middleware::getMap ( )
```

get map of key-value pairs

Returns

map of key-value pairs

4.35.3.5 getNameID()

```
std::string Middleware::getNameID ( )
```

get name id of middleware

Returns

name id

4.35.3.6 getValueFromMap()

```
bool Middleware::getValueFromMap (
             const char * key,
             std::string & value )
```

get value from map by key

Parameters

key	needed key
value	out param, if value exists should be written, do nothing otherwise

Returns

true if key exists in map, false otherwise

4.35.3.7 setContent()

```
void Middleware::setContent (
            Request * request,
            Response * response )
```

set request and response objects into Middleware

Parameters

request	request object
response	response object

4.35.4 Member Data Documentation

4.35.4.1 map

```
std::unordered_map<std::string, std::string>* Middleware::map  [protected]
```

4.35.4.2 request

```
Request* Middleware::request  [protected]
```

4.35.4.3 response

```
Response* Middleware::response  [protected]
```

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

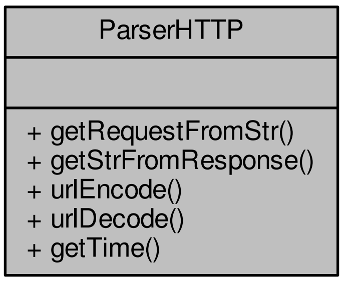
- /home/a_krava/projects/progbase3/web_server/include/middleware.h
- /home/a_krava/projects/progbase3/web_server/src/middleware.cpp

4.36 ParserHTTP Class Reference

static class for parsing, encoding, decoding any http data

```
#include <parser_http.h>
```

Collaboration diagram for ParserHTTP:



Static Public Member Functions

- static Request ∗ getRequestFromStr (std::string &str)
- static std::string getStrFromResponse (Response &response)
- static std::string urlEncode (const std::string &value)
- static std::string urlDecode (const std::string &value)
- static std::string getTime (const time_t ∗time_struct=nullptr, const char ∗format="%Y.%m.%d")

4.36.1 Detailed Description

static class for parsing, encoding, decoding any http data

ParserHTTP is used to serialize and deserialize http request, response etc.

4.36.2 Member Function Documentation

4.36.2.1 getRequestFromStr()

```
Request * ParserHTTP::getRequestFromStr (
            std::string & str )  [static]
```

Deserialize http request from input string

Parameters

str	input string

Returns

deserialized Request object

4.36.2.2 getStrFromResponse()

```
string ParserHTTP::getStrFromResponse (
            Response & response )  [static]
```

Serialize http response into string

Parameters

response	Response object

Returns

serialized string

4.36.2.3 getTime()

```
std::string ParserHTTP::getTime (
            const time_t * time_struct = nullptr,
            const char * format = "%Y.%m.%d" )  [static]
```

get date stamp in string in format from time_t

Parameters

time_struct	required time in time_t, if nullptr - execute current time
format	format of representing date in string

Returns

date stamp as string

4.36.2.4 urlDecode()

```
string ParserHTTP::urlDecode (
            const std::string & value )  [static]
```

Decode input string

Parameters

value	input string

Returns

decoded string

4.36.2.5 urlEncode()

```
string ParserHTTP::urlEncode (
            const std::string & value )  [static]
```

Encode input string

Parameters

value	input string

Returns

encoded string

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

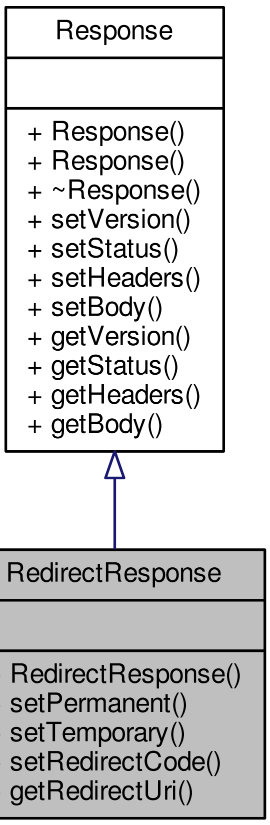
- /home/a_krava/projects/progbase3/web_server/include/parser_http.h
- /home/a_krava/projects/progbase3/web_server/src/parser_http.cpp

4.37 RedirectResponse Class Reference

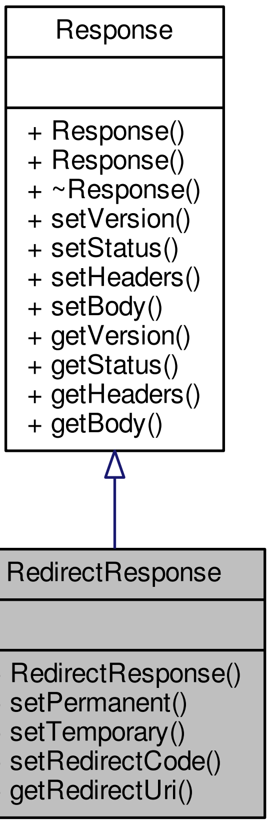
Response class which is intended to make http redirects.

```
#include <redirect_response.h>
```

Inheritance diagram for RedirectResponse:



Collaboration diagram for RedirectResponse:



Public Member Functions

- RedirectResponse (const char ∗redirectUri, const char ∗targetUri)
- void setPermanent ()
- void setTemporary ()
- void setRedirectCode (int code)
- std::string getRedirectUri ()

4.37.1 Detailed Description

Response class which is intended to make http redirects.

Inherited class RedirectResponse from Response for easiest adjusting redirects

4.37.2 Constructor & Destructor Documentation

4.37.2.1 RedirectResponse()

```
RedirectResponse::RedirectResponse (
             const char * redirectUri,
             const char * targetUri )
```

Create RedirectResponse object with redirect code 404 (you should use method to set required redirect code)

Parameters

redirectUri	input uri, which must be redirected
targetUri	destination redirect uri

4.37.3 Member Function Documentation

4.37.3.1 getRedirectUri()

```
std::string RedirectResponse::getRedirectUri ( )
```

get target uri from redirect response

Returns

destination redirect uri

4.37.3.2 setPermanent()

```
void RedirectResponse::setPermanent ( )
```

set permanent http redirect

4.37.3.3 setRedirectCode()

```
void RedirectResponse::setRedirectCode (
             int code )
```

set redirect code status

Parameters

code	http redirect code status

4.37.3.4 setTemporary()

```
void RedirectResponse::setTemporary ( )
```

set temporary http redirect

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

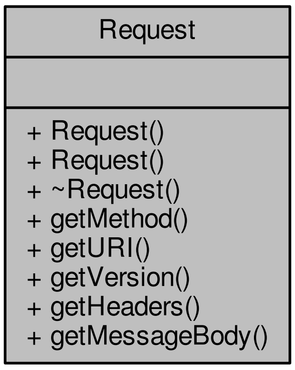
- /home/a_krava/projects/progbase3/web_server/include/redirect_response.h
- /home/a_krava/projects/progbase3/web_server/src/redirect_response.cpp

4.38 Request Class Reference

class wrapper of HTTP request

```
#include <request.h>
```

Collaboration diagram for Request:



Public Member Functions

- Request ()
- Request (HTTP::Method method, std::string &URI, HTTP::Version version, std::string &headers, std::string &body)
- ∼Request ()
- HTTP::Method getMethod ()
- URI ∗ getURI ()
- HTTP::Version getVersion ()
- Headers ∗ getHeaders ()
- MessageBody ∗ getMessageBody ()

4.38.1 Detailed Description

class wrapper of HTTP request

Object of this class is deserialized http request, where all consisting data is represented by objects of another classes

4.38.2 Constructor & Destructor Documentation

4.38.2.1 Request() [1/2]

```
Request::Request ( )
```

Makes empty Request object, where method and version is undefined

4.38.2.2 Request() [2/2]

```
Request::Request (
            HTTP::Method method,
            std::string & URI,
            HTTP::Version version,
            std::string & headers,
            std::string & body )
```

Makes Request object with declared arguments

Parameters

method	http method of request
URI	http request uri string, which is used to construct URI object
version	request http version
headers	http request headers string, which is used to construct Headers object
body	http request body string, which is used to construct MessageBody object

4.38.2.3 ∼Request()

```
Request::∼Request ( )
```

deletes URI, Headers and MessageBody objects

4.38.3 Member Function Documentation

4.38.3.1 getHeaders()

```
Headers * Request::getHeaders ( )
```

get request headers

Returns

Headers request object

4.38.3.2 getMessageBody()

```
MessageBody * Request::getMessageBody ( )
```

get request body

Returns

MessageBody request object

4.38.3.3 getMethod()

```
HTTP::Method Request::getMethod ( )
```

get request method

Returns

value of enum HTTP::Method, which represents http method.

4.38.3.4 getURI()

```
URI * Request::getURI ( )
```

get URI request object

Returns

request URI object

4.38.3.5 getVersion()

```
HTTP::Version Request::getVersion ( )
```

get http version of request

Returns

value of enum HTTP:Version, which represents http version.

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

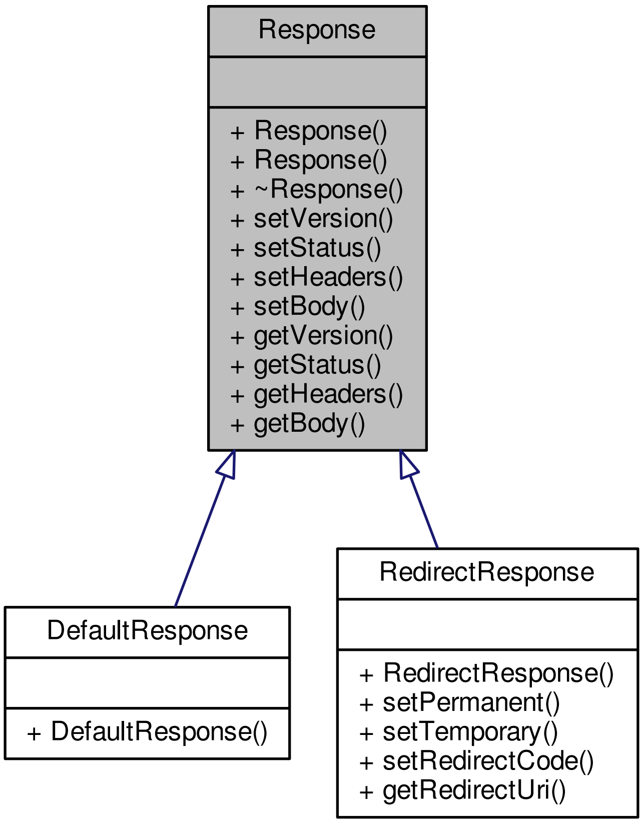
- /home/a_krava/projects/progbase3/web_server/include/request.h
- /home/a_krava/projects/progbase3/web_server/src/request.cpp

4.39 Response Class Reference

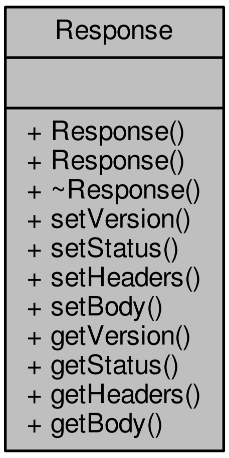
class wrapper of HTTP response

```
#include <response.h>
```

Inheritance diagram for Response:



Collaboration diagram for Response:



Public Member Functions

- Response ()
- Response (HTTP::Version version, int status, Headers &headers, MessageBody &body)
- ∼Response ()
- void setVersion (HTTP::Version version)
- void setStatus (int status)
- void setHeaders (Headers &headers)
- void setBody (MessageBody &body)
- HTTP::Version getVersion ()
- int getStatus ()
- Headers ∗ getHeaders ()
- MessageBody ∗ getBody ()

4.39.1 Detailed Description

class wrapper of HTTP response

Object of this class is representation http response

4.39.2 Constructor & Destructor Documentation

4.39.2.1 Response() [1/2]

```
Response::Response ( )
```

Create empty response with code status 501 and http version HTTP_UNDEFINED

4.39.2.2 Response() [2/2]

```
Response::Response (
            HTTP::Version version,
            int status,
            Headers & headers,
            MessageBody & body )
```

Create response and fill it with declared arguments

Parameters

version	http version of response
status	http response code status
headers	http response headers as Headers object
body	http response body as MessageBody object

4.39.2.3 ∼Response()

```
Response::∼Response ( )
```

deletes Headers and MessageBody objects

4.39.3 Member Function Documentation

4.39.3.1 getBody()

```
MessageBody * Response::getBody ( )
```

get body of http response

Returns

http response body as MessageBody object

4.39.3.2 getHeaders()

```
Headers * Response::getHeaders ( )
```

get headers of http response

Returns

http response headers as Headers object

4.39.3.3 getStatus()

```
int Response::getStatus ( )
```

get http code status of response

Returns

http response code status

4.39.3.4 getVersion()

```
HTTP::Version Response::getVersion ( )
```

get http version of response

Returns

http response version

4.39.3.5 setBody()

```
void Response::setBody (
            MessageBody & body )
```

Set to response MessageBody object and deleting previous one

Parameters

body	MessageBody object

4.39.3.6 setHeaders()

```
void Response::setHeaders (
            Headers & headers )
```

Set to response Headers object and deleting previous one

Parameters

headers	headers object

4.39.3.7 setStatus()

```
void Response::setStatus (
            int status )
```

Set to response http status

Parameters

status	http code status

4.39.3.8 setVersion()

```
void Response::setVersion (
            HTTP::Version version )
```

Set to response http version

Parameters

version	value of enum HTTP::Version, which represents http version.

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

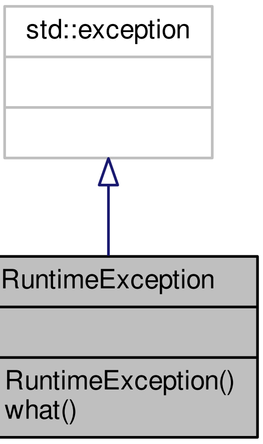
- /home/a_krava/projects/progbase3/web_server/include/response.h
- /home/a_krava/projects/progbase3/web_server/src/response.cpp

4.40 RuntimeException Class Reference

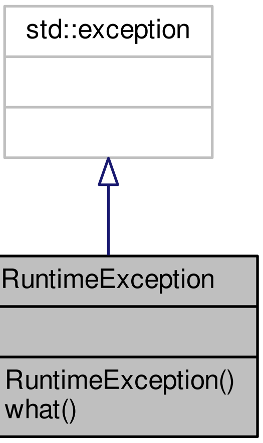
exception class for program errors

```
#include <runtime_exception.h>
```

Inheritance diagram for RuntimeException:



Collaboration diagram for RuntimeException:



Public Member Functions

- RuntimeException (const std::string &error)
- const char ∗ what () const noexcept override

4.40.1 Detailed Description

exception class for program errors

Inherited class from std::exception

4.40.2 Constructor & Destructor Documentation

4.40.2.1 RuntimeException()

```
RuntimeException::RuntimeException (
            const std::string & error )  [inline], [explicit]
```

create RuntimeException with error explanation

Parameters

error	explanation of thrown error

4.40.3 Member Function Documentation

4.40.3.1 what()

```
const char* RuntimeException::what ( ) const  [inline], [override], [noexcept]
```

get error information

Returns

error info as string

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

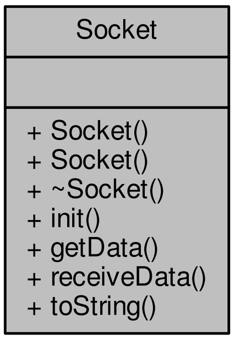
- /home/a_krava/projects/progbase3/web_server/include/runtime_exception.h

4.41 Socket Class Reference

wrapper functions to send/receive data via web-sockets

```
#include <socket.h>
```

Collaboration diagram for Socket:



Public Member Functions

- Socket (InitParams params)
- Socket (std::string ip, int port, bool isIPv6)
- ∼Socket ()
- void init ()
- std::string getData ()
- void receiveData (std::string &data)
- std::string toString ()

4.41.1 Detailed Description

wrapper functions to send/receive data via web-sockets

Implementation of sockets function for Linux OS

4.41.2 Constructor & Destructor Documentation

4.41.2.1 Socket() [1/2]

```
Socket::Socket (
            InitParams params )
```

Filling object with host address information represented by InitParams object

Parameters

params	object of class InitParams with info about host address

4.41.2.2 Socket() [2/2]

```
Socket::Socket (
              std::string ip,
              int port,
              bool isIPv6 )
```

Filling object with host address information to do method init in future

Parameters

ip	host address ip as string (IPv4 or IPv6)
port	host address port
isIPv6	if argument ip is version 6 set true, false otherwise

4.41.2.3 ∼Socket()

```
Socket::∼Socket ( )
```

closing opened host socket

4.41.3 Member Function Documentation

4.41.3.1 getData()

```
std::string Socket::getData ( )
```

Accepting all clients at configured host and reading data from current client

Exceptions

RuntimeException	when got error with reading from client

Returns

data as string

4.41.3.2 init()

```
void Socket::init ( )
```

create socket, binding created socket to host address and start listening port

Exceptions

RuntimeException	when address was invalid, busy etc.

4.41.3.3 receiveData()

```
void Socket::receiveData (
             std::string & data )
```

receive data to client, which sent data before it (method getData was used)

Parameters

data	serialized data into string

4.41.3.4 toString()

```
std::string Socket::toString ( )
```

get information about current ip and port of host

Returns

string in format "ip: xxx.xxx.xxx.xxx port: xx"

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

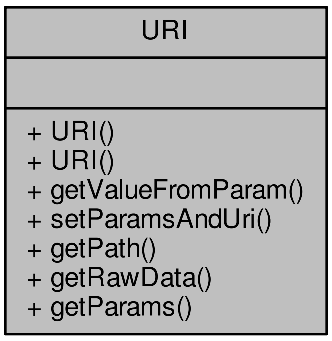
- /home/a_krava/projects/progbase3/web_server/include/socket.h
- /home/a_krava/projects/progbase3/web_server/src/socket.cpp

4.42 URI Class Reference

class represents http uri

```
#include <uri.h>
```

Collaboration diagram for URI:



Public Member Functions

- URI ()
- URI (std::string &uri)
- bool getValueFromParam (const char ∗key, std::string &value)
- void setParamsAndUri (std::string &uri)
- std::string getPath ()
- std::string getRawData ()
- std::unordered_map< std::string, std::string > getParams ()

4.42.1 Detailed Description

class represents http uri

URI consist of uri - a string of uri without arguments, and map of key-value pairs which are deserialized parameters of uri

4.42.2 Constructor & Destructor Documentation

4.42.2.1 URI() [1/2]

```
URI::URI ( )
```

Create empty URI object

4.42.2.2 URI() [2/2]

```
URI::URI (
            std::string & uri )
```

Construct URI object from string, deserialize all params and decode input string

Parameters

uri	http uri string

4.42.3 Member Function Documentation

4.42.3.1 getParams()

```
unordered_map< string, string > URI::getParams ( )
```

get deserialized and decoded map of http uri params

Returns

map of key-value pairs from uri params

4.42.3.2 getPath()

```
string URI::getPath ( )
```

get decoded uri path without params

Returns

http uri string without params

4.42.3.3 getRawData()

```
std::string URI::getRawData ( )
```

get unchanged uri string with params

Returns

http uri string

4.42.3.4 getValueFromParam()

```
bool URI::getValueFromParam (
             const char * key,
             std::string & value )
```

get value by key from uri

Parameters

key	key in params of http uri
value	out param, where will be written value if exists or do nothing otherwise

Returns

true if key exists in map, false otherwise

4.42.3.5 setParamsAndUri()

```
void URI::setParamsAndUri (
            std::string & uri )
```

deserialize and decode input http uri into path and params

Parameters

uri	http uri string

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

- /home/a_krava/projects/progbase3/web_server/include/uri.h
- /home/a_krava/projects/progbase3/web_server/src/uri.cpp

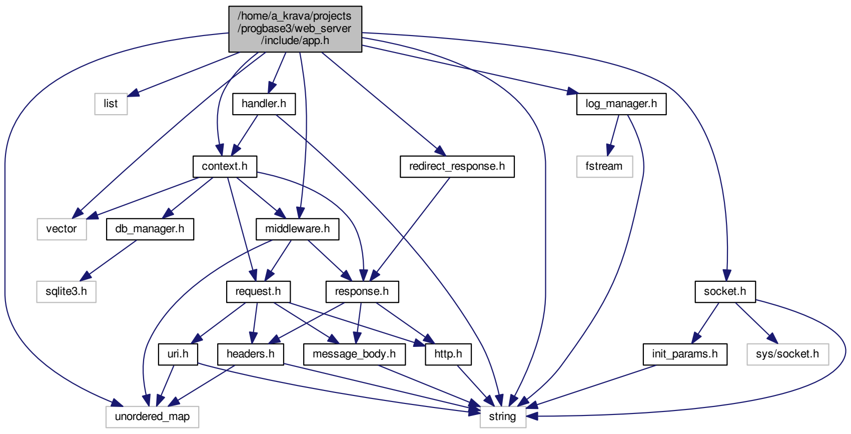
Chapter 5

File Documentation

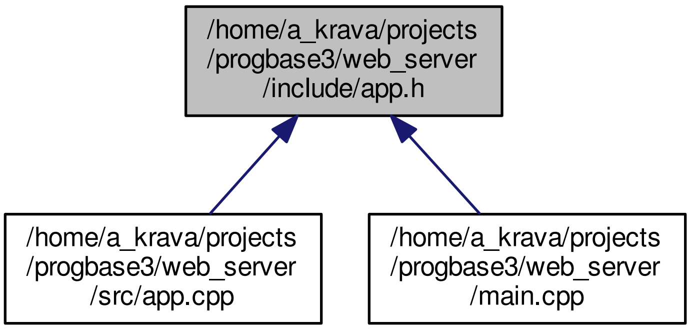
5.1 /home/a_krava/projects/progbase3/web_server/include/app.h File Reference

```
#include <string>
#include <list>
#include <vector>
#include <unordered_map>
#include <handler.h>
#include <socket.h>
#include <redirect_response.h>
#include <log_manager.h>
#include <middleware.h>
#include <context.h>
```

Include dependency graph for app.h:



This graph shows which files directly or indirectly include this file:



Classes

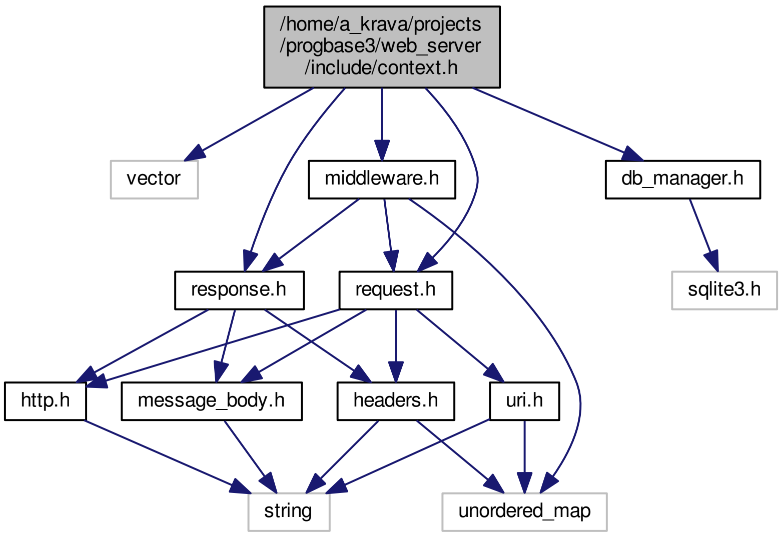
- class App

  The main class of the framework. Each object of this class is an independent web-application, which could be configured by handlers, middleware etc.

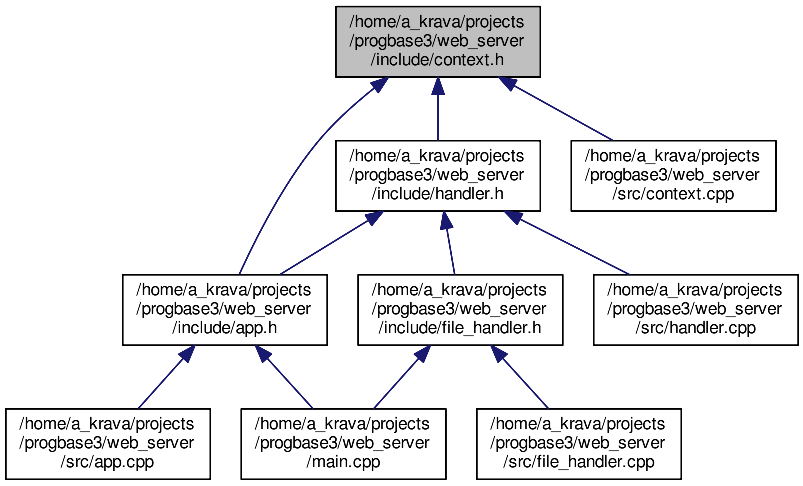
5.2 /home/a_krava/projects/progbase3/web_server/include/context.h File Reference

```
#include <vector>
#include <request.h>
#include <response.h>
#include <middleware.h>
#include <db_manager.h>
```

Include dependency graph for context.h:



This graph shows which files directly or indirectly include this file:



Classes

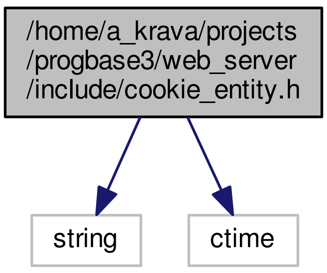
- class Context

  This class is wrapper for important data (like Response, DB, Middleware etc.), which is needed to handlers.

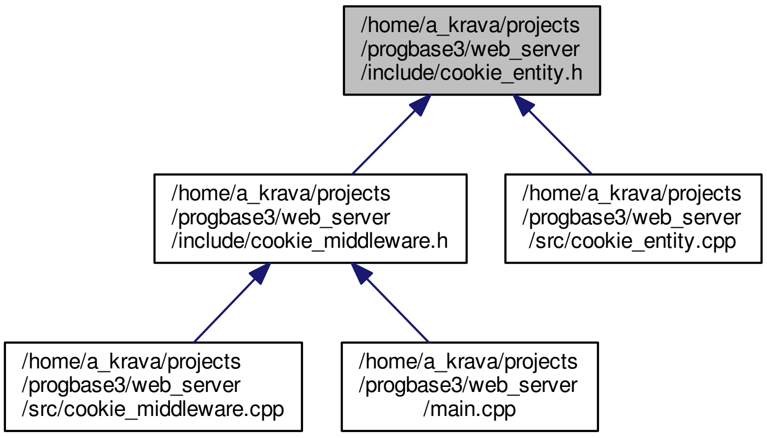
5.3 /home/a_krava/projects/progbase3/web_server/include/cookie_entity.h File Reference

```
#include <string>
#include <ctime>
```

Include dependency graph for cookie_entity.h:



This graph shows which files directly or indirectly include this file:



Classes

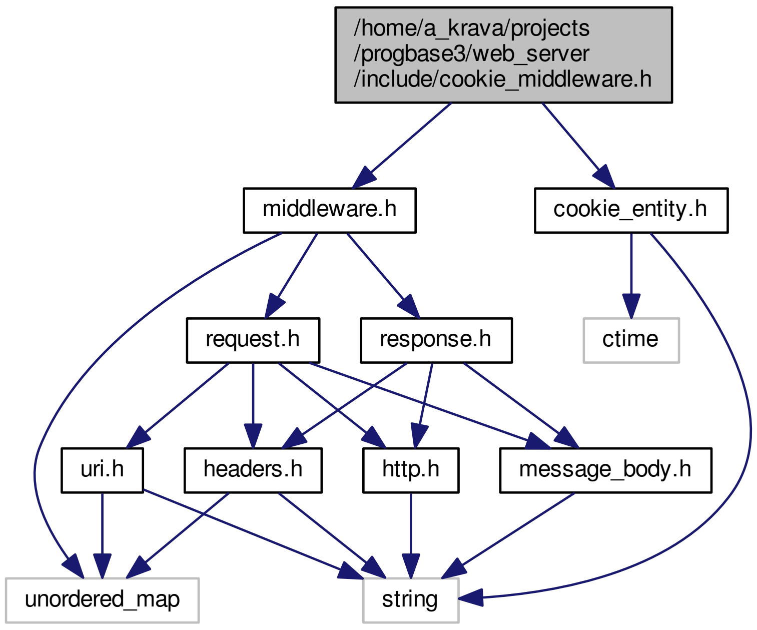
- class CookieEntity

  Class wrapper for Cookies. Allow you adjust parameters od each http cookie. Used by CookieMiddleware.

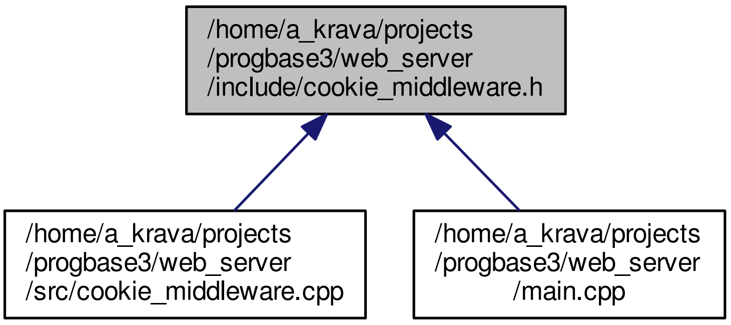
5.4 /home/a_krava/projects/progbase3/web_server/include/cookie_middleware.h File Reference

```
#include <middleware.h>
#include <cookie_entity.h>
```

Include dependency graph for cookie_middleware.h:



This graph shows which files directly or indirectly include this file:


/home/a_krava/projects /progbase3/web_server /include/cookie_middleware.h
/home/a_krava/projects /progbase3/web_server /src/cookie_middleware.cpp
/home/a_krava/projects /progbase3/web_server /main.cpp

Classes

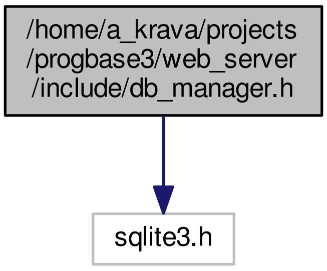
- class CookieMiddleware

  inherited class to parse cookie from http request

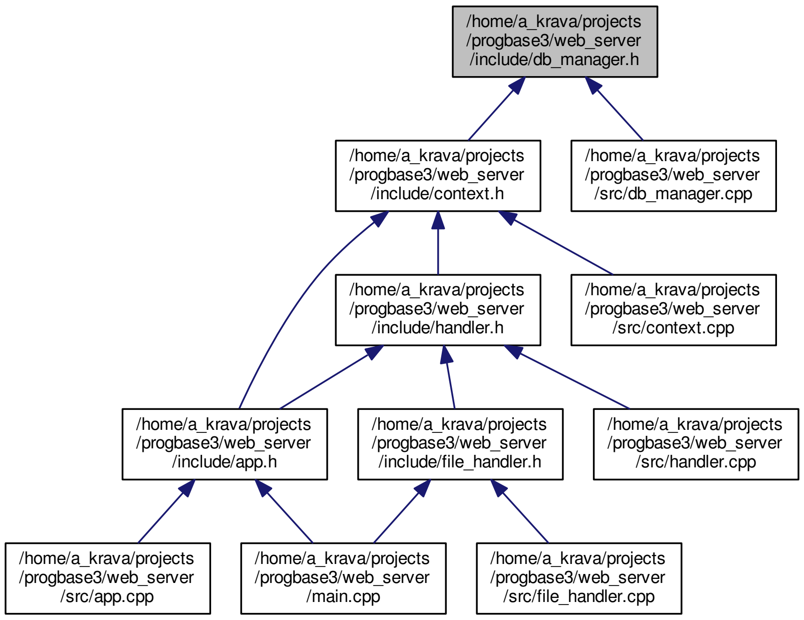
5.5 /home/a_krava/projects/progbase3/web_server/include/db_manager.h File Reference

```
#include <sqlite3.h>
```

Include dependency graph for db_manager.h:



This graph shows which files directly or indirectly include this file:



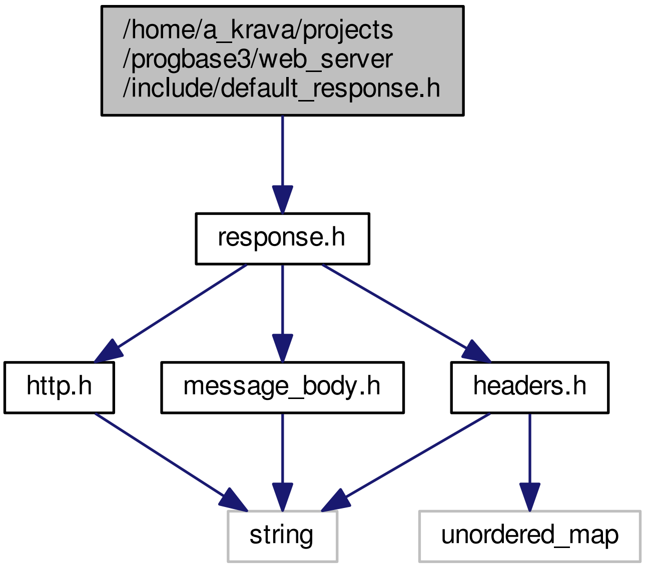
Classes

- class DBManager

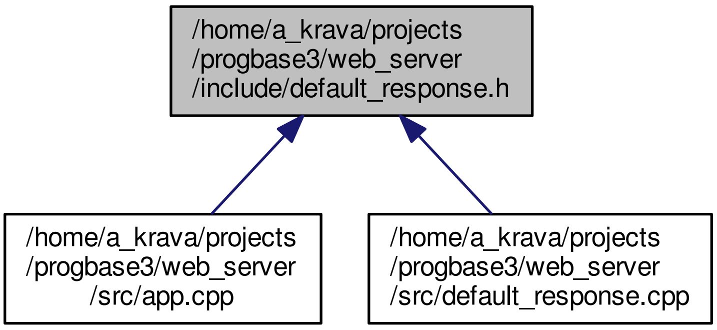
  allow perform sql queries to db

5.6 /home/a_krava/projects/progbase3/web_server/include/default_response.h File Reference

`#include <response.h>`
Include dependency graph for default_response.h:



This graph shows which files directly or indirectly include this file:



Classes

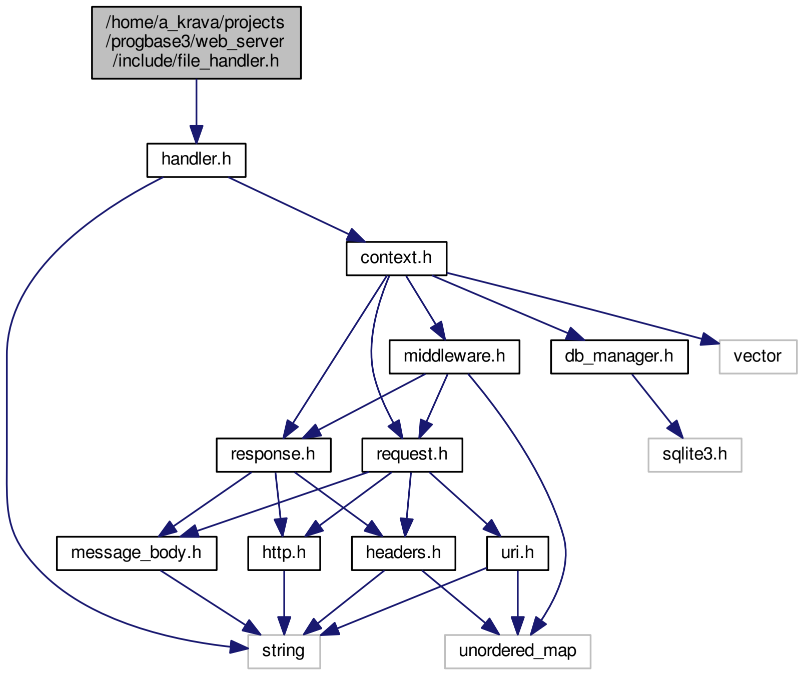
- class DefaultResponse

  Response class which is intended to make sample html pages on status codes.

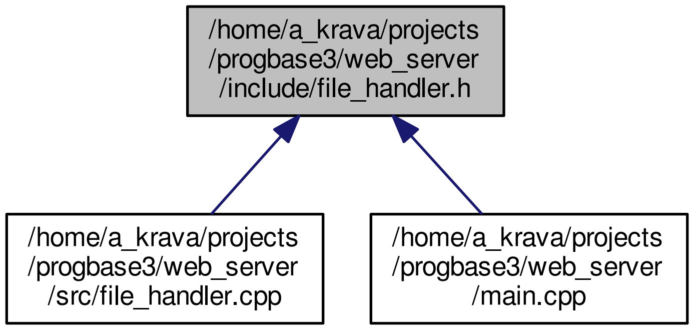
5.7 /home/a_krava/projects/progbase3/web_server/include/file_handler.h File Reference

```
#include <handler.h>
```

Include dependency graph for file_handler.h:



This graph shows which files directly or indirectly include this file:


/home/a_krava/projects/progbase3/web_server/include/file_handler.h
/home/a_krava/projects/progbase3/web_server/src/file_handler.cpp
/home/a_krava/projects/progbase3/web_server/main.cpp

Classes

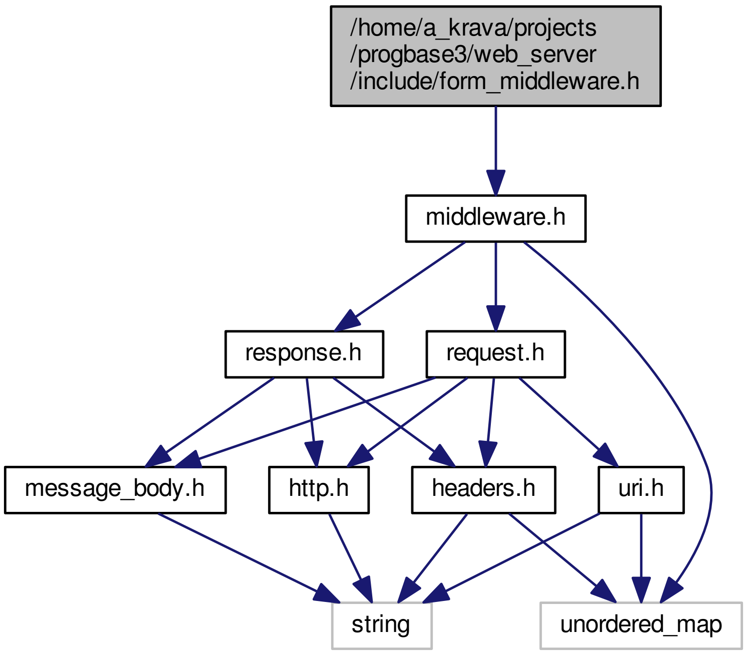
- class FileHandler

  this class allow you to set any file of filesystem as response body

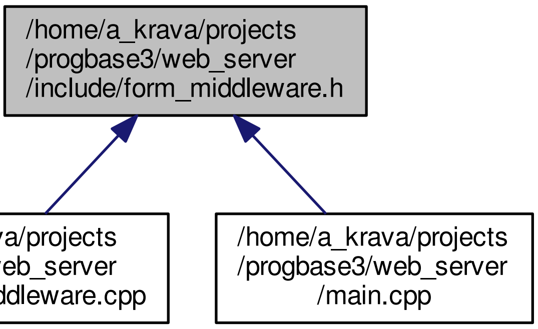
5.8 /home/a_krava/projects/progbase3/web_server/include/form_middleware.h File Reference

```
#include <middleware.h>
```

Include dependency graph for form_middleware.h:



This graph shows which files directly or indirectly include this file:


/home/a_krava/projects /progbase3/web_server /include/form_middleware.h
/home/a_krava/projects /progbase3/web_server /src/form_middleware.cpp
/home/a_krava/projects /progbase3/web_server /main.cpp

Classes

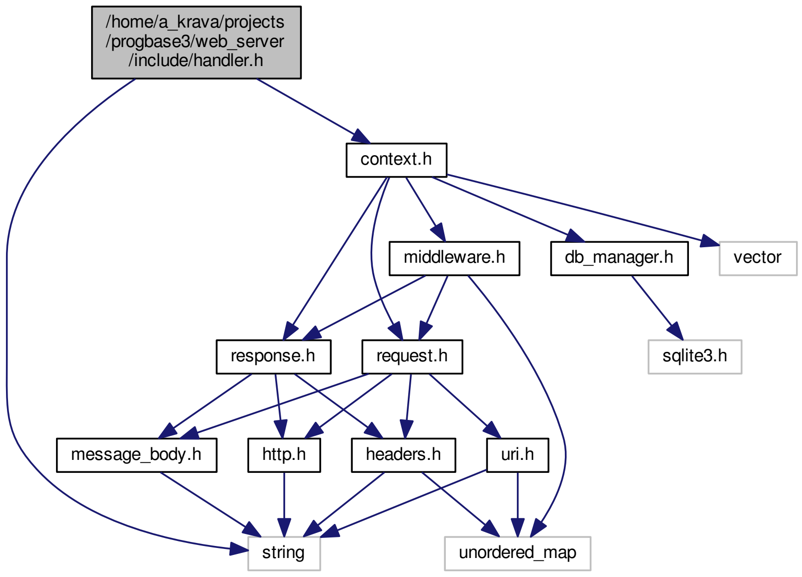
- class FormMiddleware

  inherited class to parse application/x-www-form-urlencoded

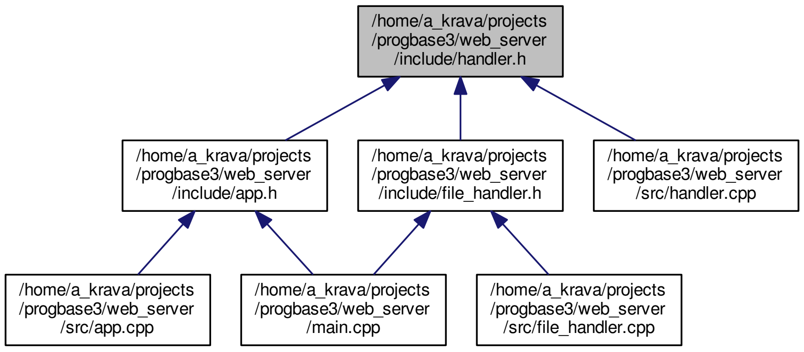
5.9 /home/a_krava/projects/progbase3/web_server/include/handler.h File Reference

```
#include <string>
#include <context.h>
```

Include dependency graph for handler.h:



This graph shows which files directly or indirectly include this file:


/home/a_krava/projects/progbase3/web_server/include/handler.h
/home/a_krava/projects/progbase3/web_server/include/app.h
/home/a_krava/projects/progbase3/web_server/include/file_handler.h
/home/a_krava/projects/progbase3/web_server/src/handler.cpp
/home/a_krava/projects/progbase3/web_server/src/app.cpp
/home/a_krava/projects/progbase3/web_server/main.cpp
/home/a_krava/projects/progbase3/web_server/src/file_handler.cpp

Classes

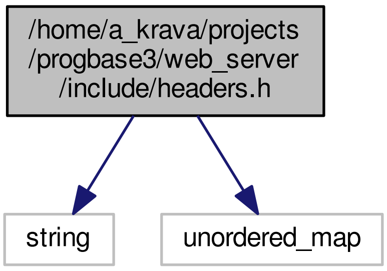
- class Handler

  object of this class executes every time on new request, this object (and others) construct response to client

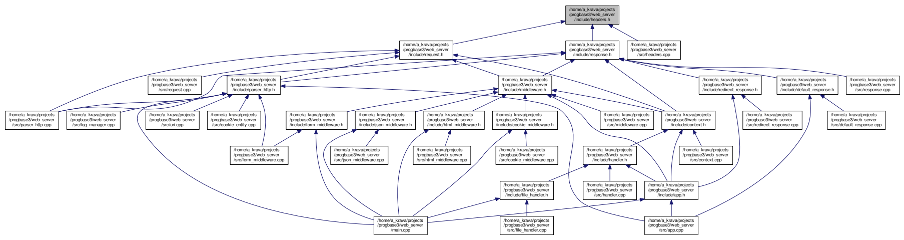
5.10 /home/a_krava/projects/progbase3/web_server/include/headers.h File Reference

```
#include <string>
#include <unordered_map>
```

Include dependency graph for headers.h:


/home/a_krava/projects/progbase3/web_server/include/headers.h
string
unordered_map

This graph shows which files directly or indirectly include this file:



Classes

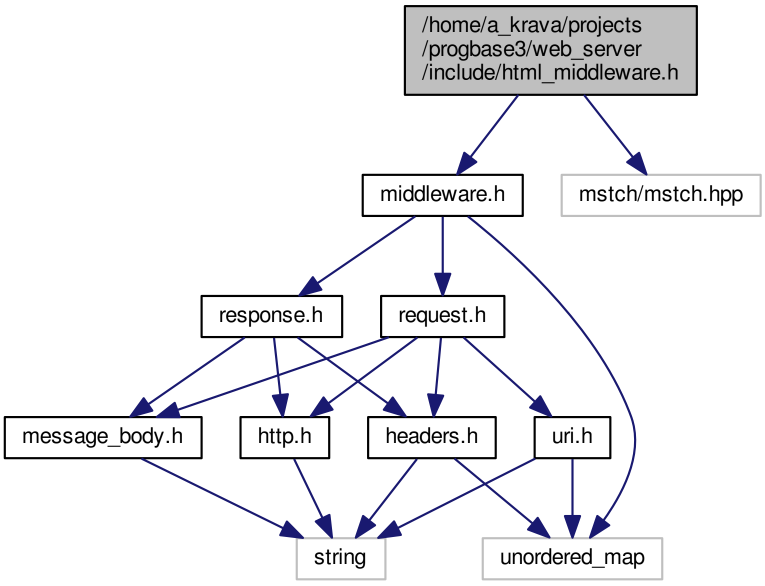
- class Headers

    wrapper class for http headers

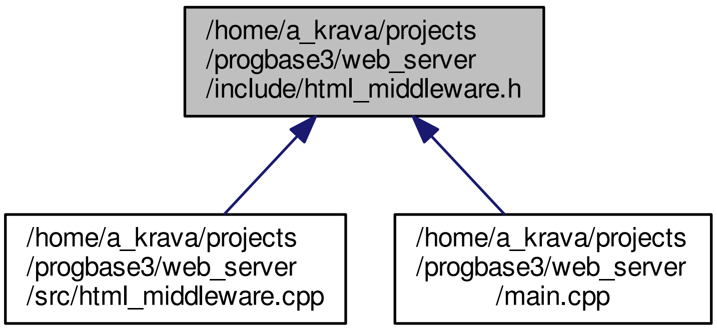
5.11 /home/a_krava/projects/progbase3/web_server/include/html_middleware.h File Reference

```
#include <middleware.h>
#include <mstch/mstch.hpp>
```

Include dependency graph for html_middleware.h:



This graph shows which files directly or indirectly include this file:



Classes

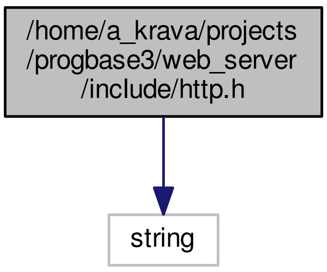
- class HtmlMiddleware

  inherited class to render html pages from templates

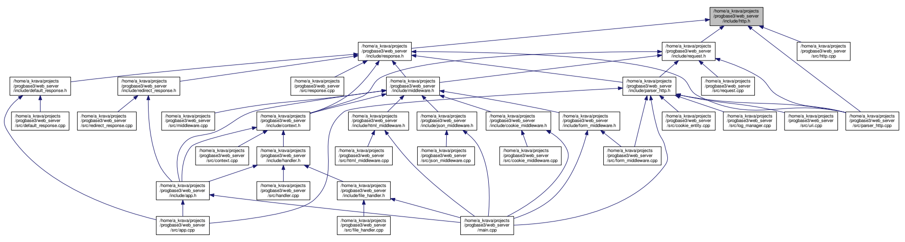
5.12 /home/a_krava/projects/progbase3/web_server/include/http.h File Reference

```
#include <string>
```

Include dependency graph for http.h:



This graph shows which files directly or indirectly include this file:



Classes

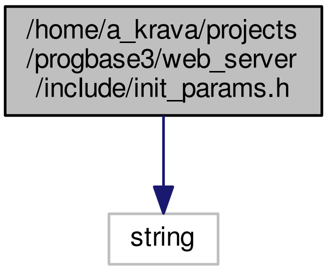
- class HTTP

  static class describes http method, version, and allow to convert it from/to string/enumeration

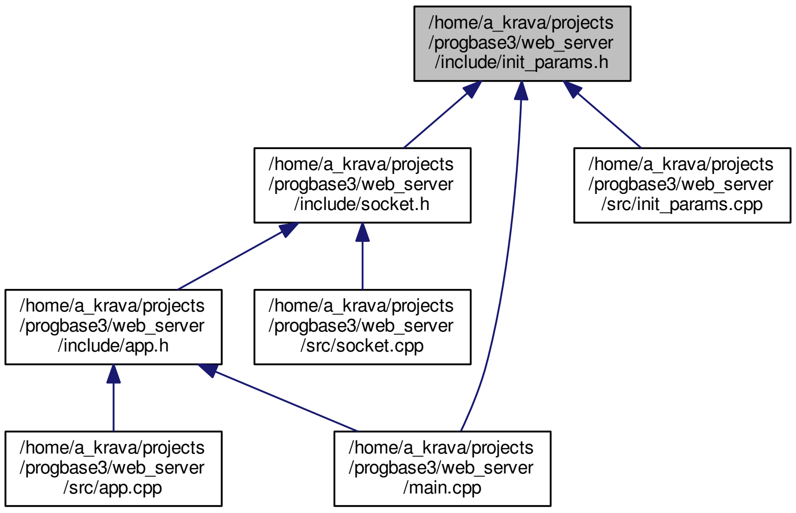
5.13 /home/a_krava/projects/progbase3/web_server/include/init_params.h File Reference

```
#include <string>
```

Include dependency graph for init_params.h:



This graph shows which files directly or indirectly include this file:


/home/a_krava/projects /progbase3/web_server /include/init_params.h
/home/a_krava/projects /progbase3/web_server /include/socket.h
/home/a_krava/projects /progbase3/web_server /src/init_params.cpp
/home/a_krava/projects /progbase3/web_server /include/app.h
/home/a_krava/projects /progbase3/web_server /src/socket.cpp
/home/a_krava/projects /progbase3/web_server /src/app.cpp
/home/a_krava/projects /progbase3/web_server /main.cpp

Classes

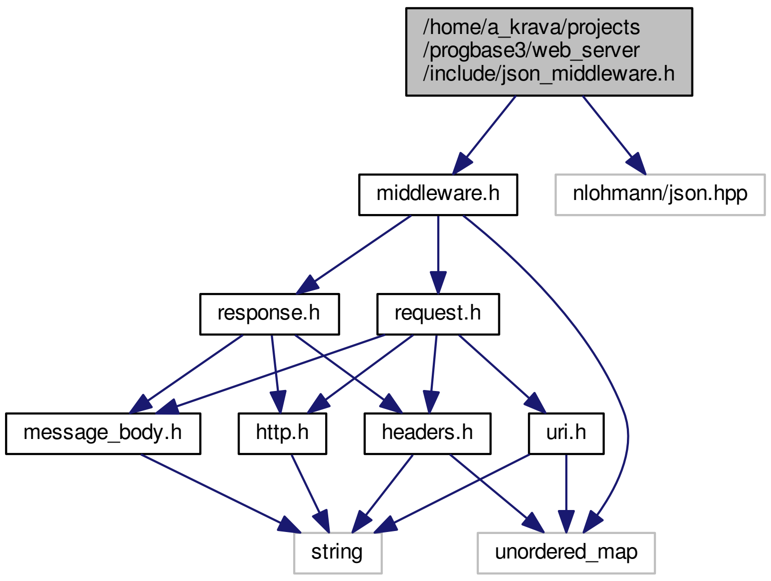
- class InitParams

  InitParams is intended to get web-server configs from command line arguments.

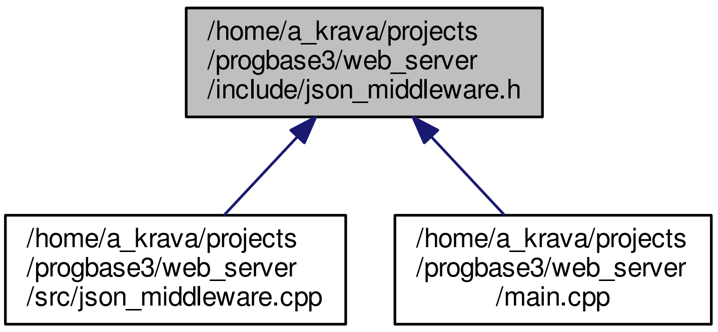
5.14 /home/a_krava/projects/progbase3/web_server/include/json_middleware.h File Reference

```
#include <middleware.h>
#include <nlohmann/json.hpp>
```

Include dependency graph for json_middleware.h:



This graph shows which files directly or indirectly include this file:



Classes

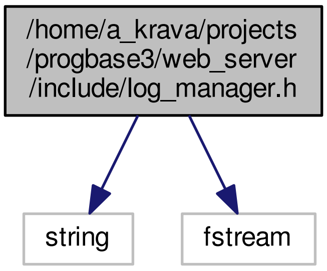
- class JsonMiddleware

  inherited class to perform any actions with json data

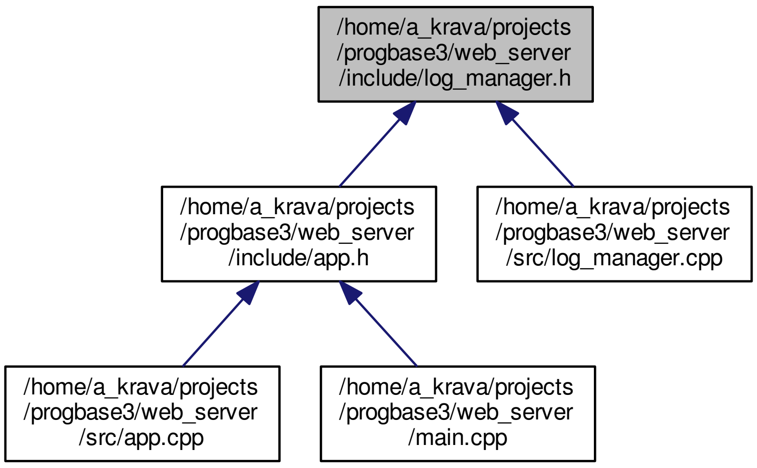
5.15 /home/a_krava/projects/progbase3/web_server/include/log_manager.h File Reference

```
#include <string>
#include <fstream>
```

Include dependency graph for log_manager.h:



This graph shows which files directly or indirectly include this file:



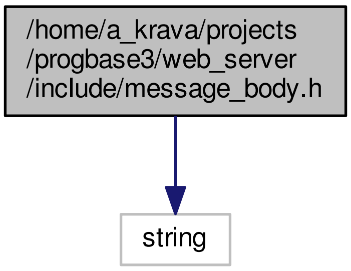
Classes

- class LogManager

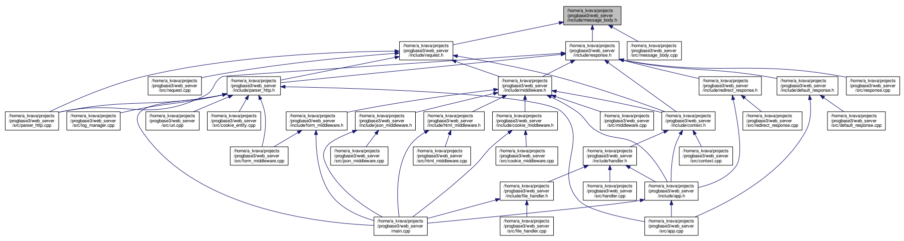
  logging info into file

5.16 /home/a_krava/projects/progbase3/web_server/include/message_body.h File Reference

```
#include <string>
```
Include dependency graph for message_body.h:



This graph shows which files directly or indirectly include this file:



Classes

- class MessageBody

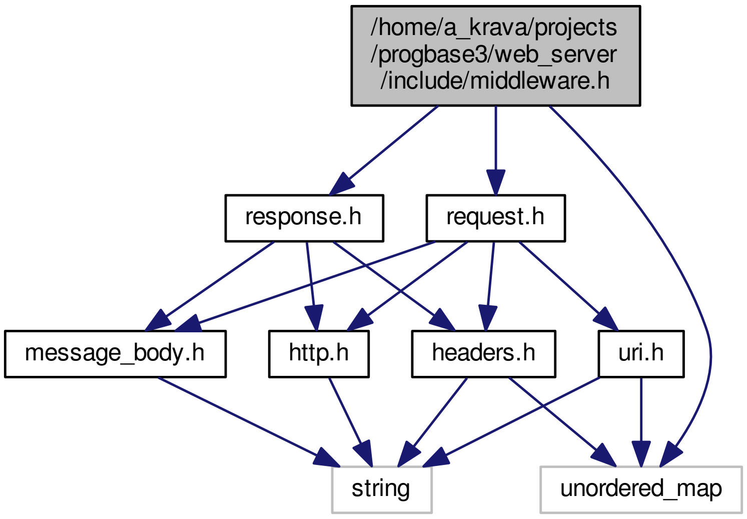
  wrapper class for http body

5.17 /home/a_krava/projects/progbase3/web_server/include/middleware.h File Reference

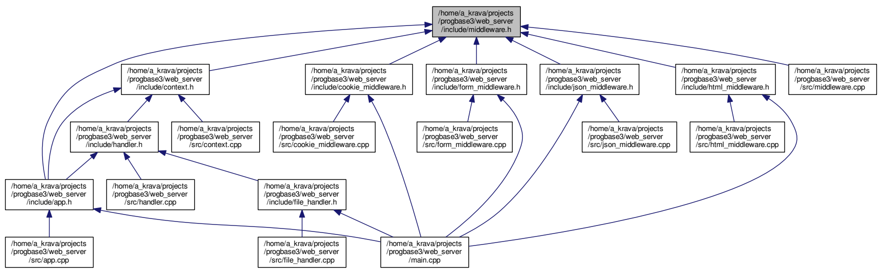
```
#include <response.h>
#include <unordered_map>
```

```
#include <request.h>
```

Include dependency graph for middleware.h:



This graph shows which files directly or indirectly include this file:



Classes

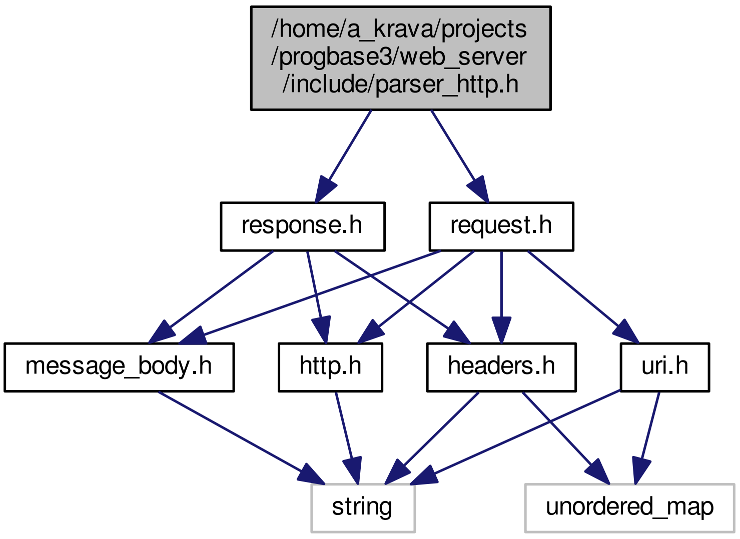
- class Middleware

  class wrapper for middleware

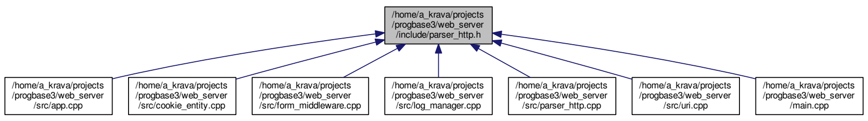
5.18 /home/a_krava/projects/progbase3/web_server/include/parser_http.h File Reference

```
#include <request.h>
#include <response.h>
```

Include dependency graph for parser_http.h:



This graph shows which files directly or indirectly include this file:



Classes

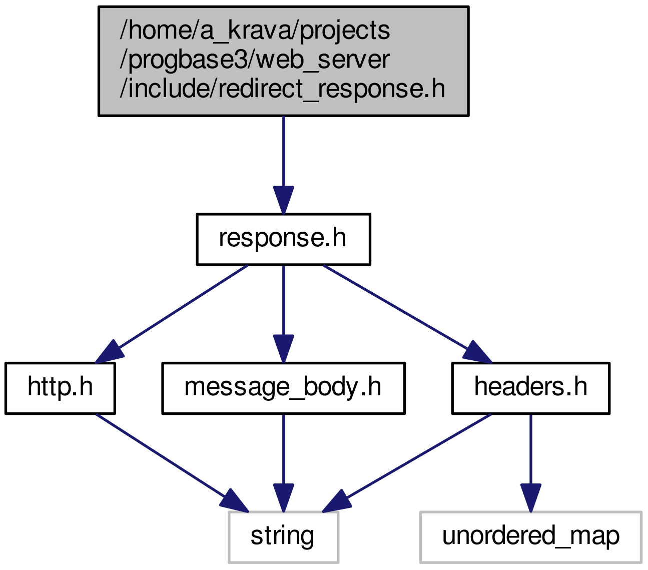
- class ParserHTTP

  static class for parsing, encoding, decoding any http data

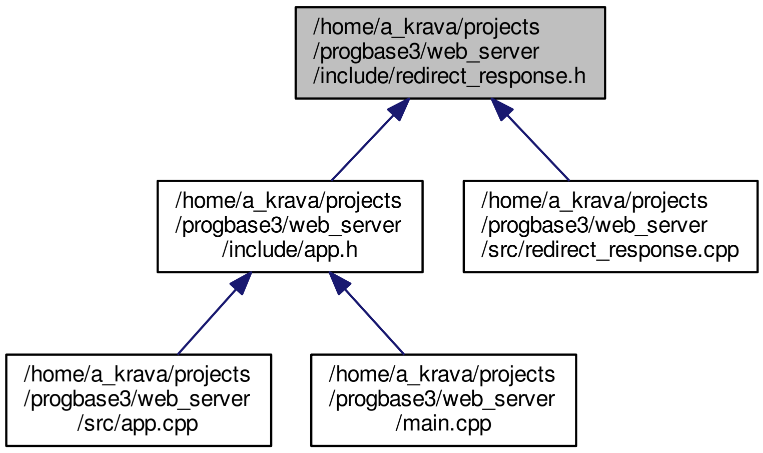
5.19 /home/a_krava/projects/progbase3/web_server/include/redirect_response.h File Reference

```
#include <response.h>
```

Include dependency graph for redirect_response.h:



This graph shows which files directly or indirectly include this file:



Classes

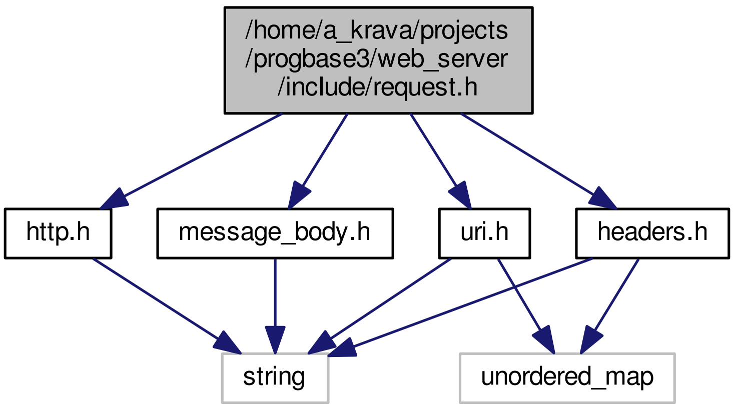
- class RedirectResponse

  Response class which is intended to make http redirects.

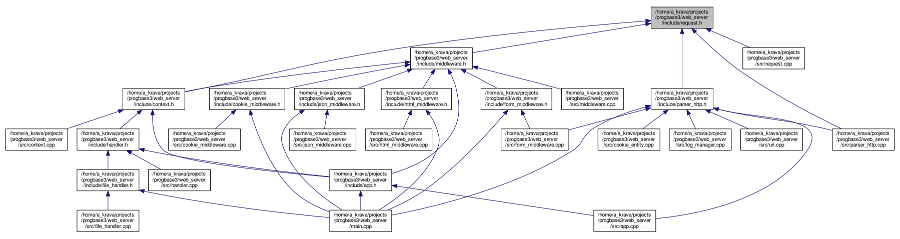
5.20 /home/a_krava/projects/progbase3/web_server/include/request.h File Reference

```
#include <http.h>
#include <message_body.h>
#include <uri.h>
#include <headers.h>
```

Include dependency graph for request.h:



This graph shows which files directly or indirectly include this file:



Classes

- class Request

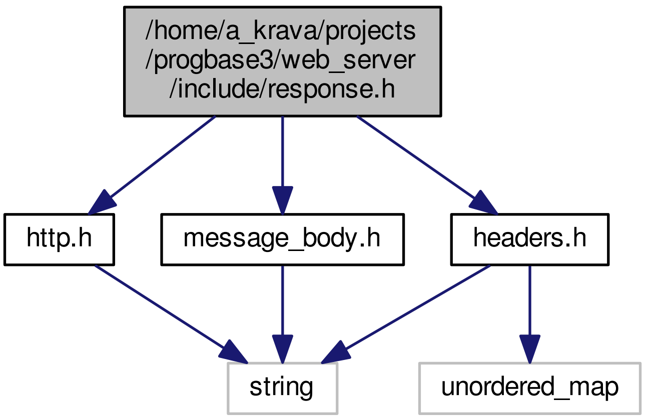
    class wrapper of HTTP request

5.21 /home/a_krava/projects/progbase3/web_server/include/response.h File Reference

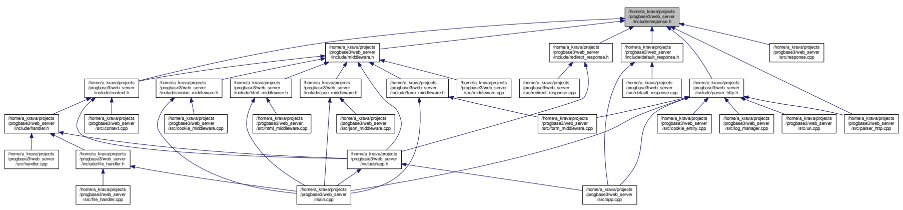
```
#include <http.h>
#include <headers.h>
```

```
#include <message_body.h>
```

Include dependency graph for response.h:



This graph shows which files directly or indirectly include this file:



Classes

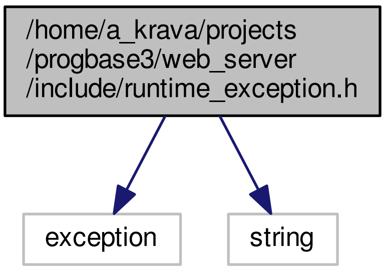
- class Response

  class wrapper of HTTP response

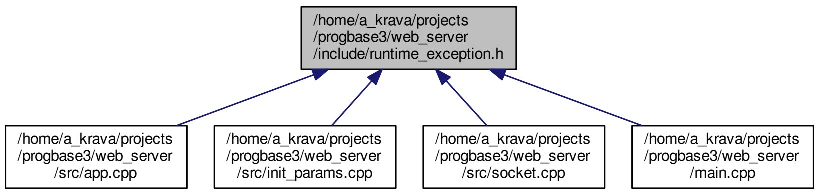
5.22 /home/a_krava/projects/progbase3/web_server/include/runtime_exception.h File Reference

```
#include <exception>
#include <string>
```

Include dependency graph for runtime_exception.h:



This graph shows which files directly or indirectly include this file:



Classes

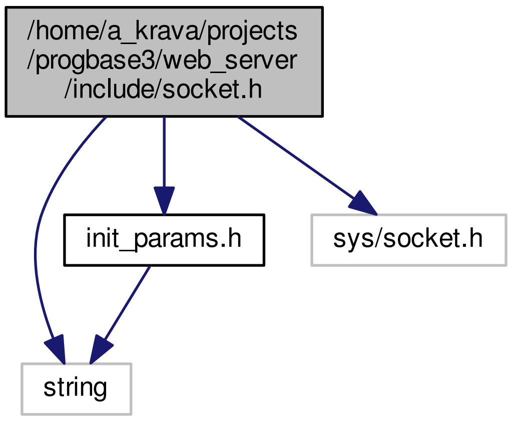
- class RuntimeException

  exception class for program errors

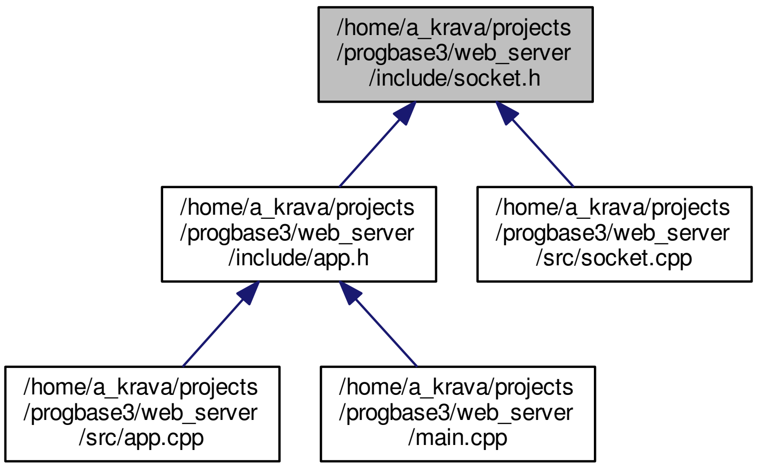
5.23 /home/a_krava/projects/progbase3/web_server/include/socket.h File Reference

```
#include <string>
#include <init_params.h>
#include <sys/socket.h>
```

Include dependency graph for socket.h:



This graph shows which files directly or indirectly include this file:



Classes

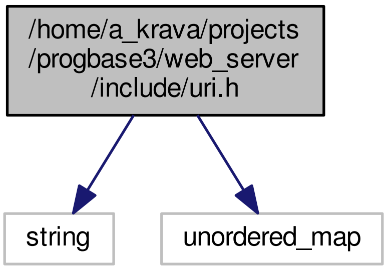
- class Socket

  wrapper functions to send/receive data via web-sockets

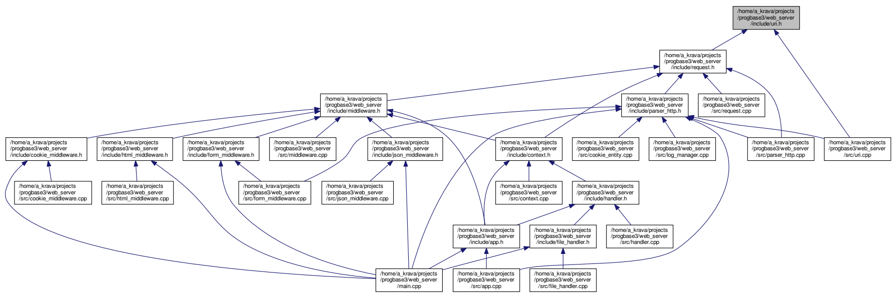
5.24 /home/a_krava/projects/progbase3/web_server/include/uri.h File Reference

```
#include <string>
#include <unordered_map>
```

Include dependency graph for uri.h:



This graph shows which files directly or indirectly include this file:



Classes

- class URI

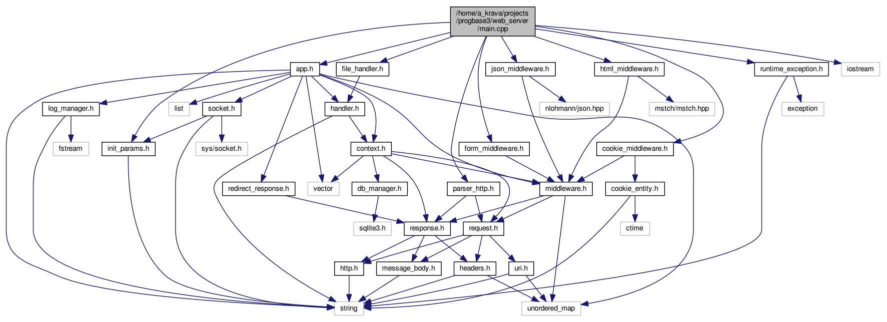
  class represents http uri

5.25 /home/a_krava/projects/progbase3/web_server/main.cpp File Reference

```
#include <init_params.h>
#include <iostream>
#include <app.h>
#include <json_middleware.h>
#include <cookie_middleware.h>
#include <form_middleware.h>
#include <html_middleware.h>
#include <file_handler.h>
```

```
#include <runtime_exception.h>
#include <parser_http.h>
```

Include dependency graph for main.cpp:



Classes

- class HandlerCommon
- class HandlerTemplate
- class HandlerCookie
- class HandlerRenderTemplate
- class HandlerIndex
- class HandlerTrack
- class HandlerCalculate
- class HandlerCalculatePost
- class HandlerEstimate
- class HandlerEstimatePost
- class HandlerMap
- class HandlerOrder
- class HandlerOrderPost
- class HandlerCommonInfo
- class HandlerNews
- class HandlerFeedback
- class HandlerFeedbackPost
- class HandlerApi

Functions

- int main (int argc, char ∗∗argv)

5.25.1 Function Documentation

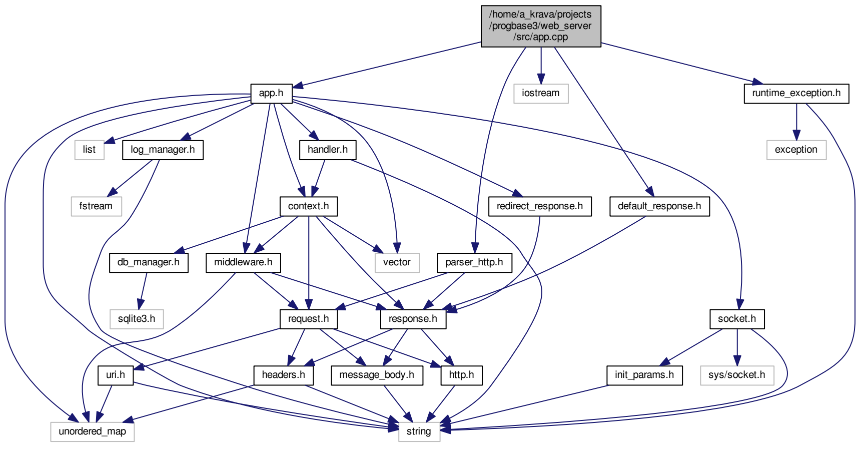
5.25.1.1 main()

```
int main (
            int argc,
            char ** argv )
```

5.26 /home/a_krava/projects/progbase3/web_server/src/app.cpp File Reference

```
#include <app.h>
#include <runtime_exception.h>
#include <iostream>
#include <parser_http.h>
#include <default_response.h>
```

Include dependency graph for app.cpp:



Macros

- #define __DB "./../db/db_file"

5.26.1 Macro Definition Documentation

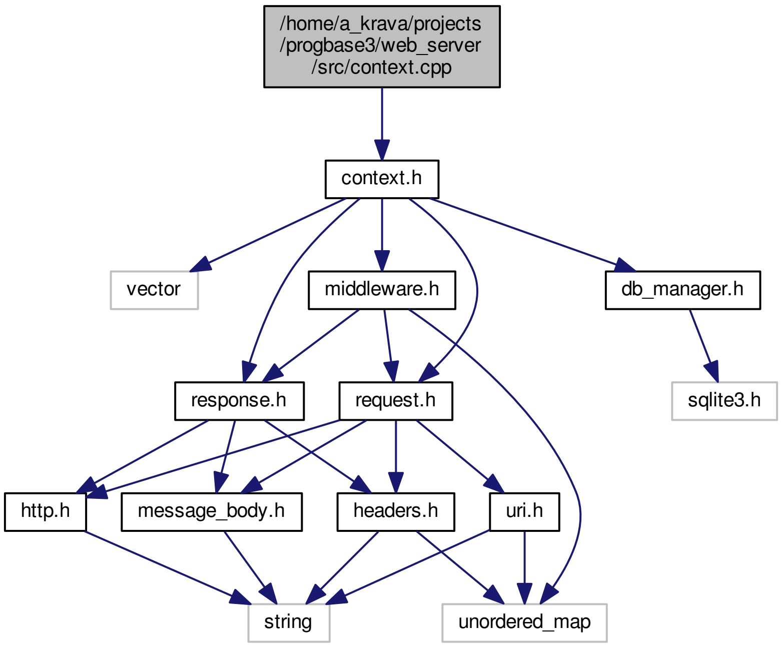
5.26.1.1 __DB

```
#define __DB "./../db/db_file"
```

5.27 /home/a_krava/projects/progbase3/web_server/src/context.cpp File Reference

```
#include <context.h>
```

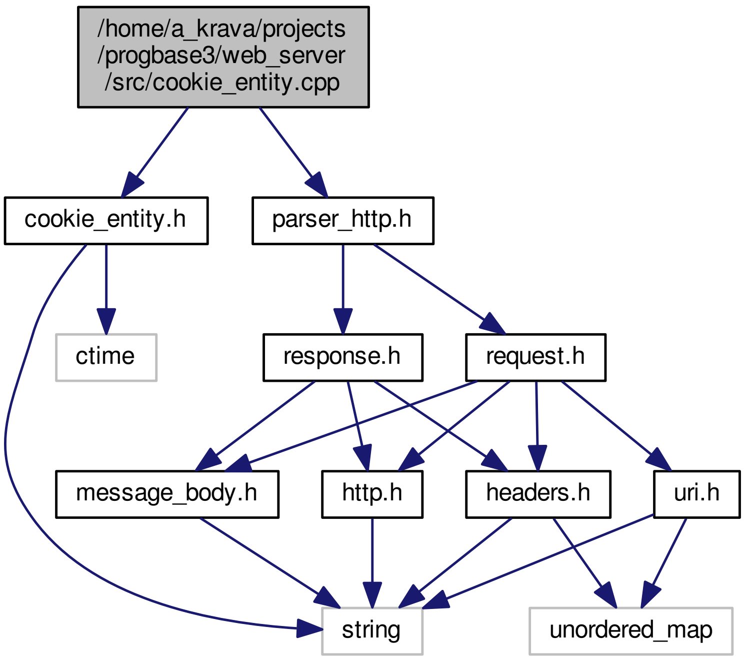
Include dependency graph for context.cpp:



5.28 /home/a_krava/projects/progbase3/web_server/src/cookie_entity.cpp File Reference

```
#include <cookie_entity.h>
#include <parser_http.h>
```

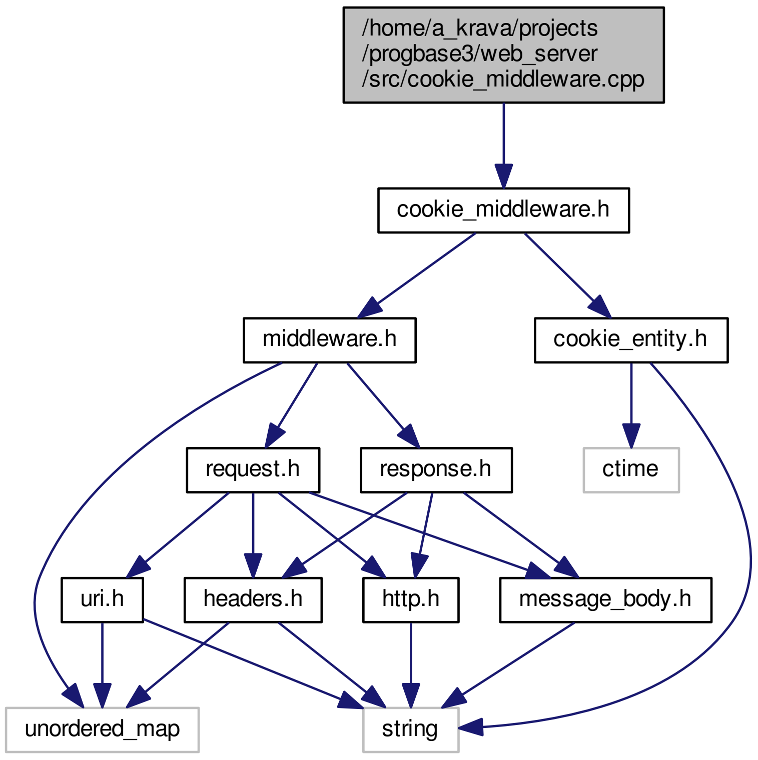
Include dependency graph for cookie_entity.cpp:



5.29 /home/a_krava/projects/progbase3/web_server/src/cookie_middleware.cpp File Reference

```
#include <cookie_middleware.h>
```

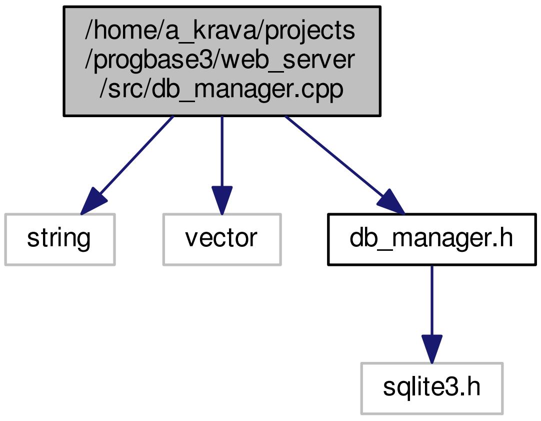
Include dependency graph for cookie_middleware.cpp:



5.30 /home/a_krava/projects/progbase3/web_server/src/db_manager.cpp File Reference

```
#include <string>
#include <vector>
#include <db_manager.h>
```

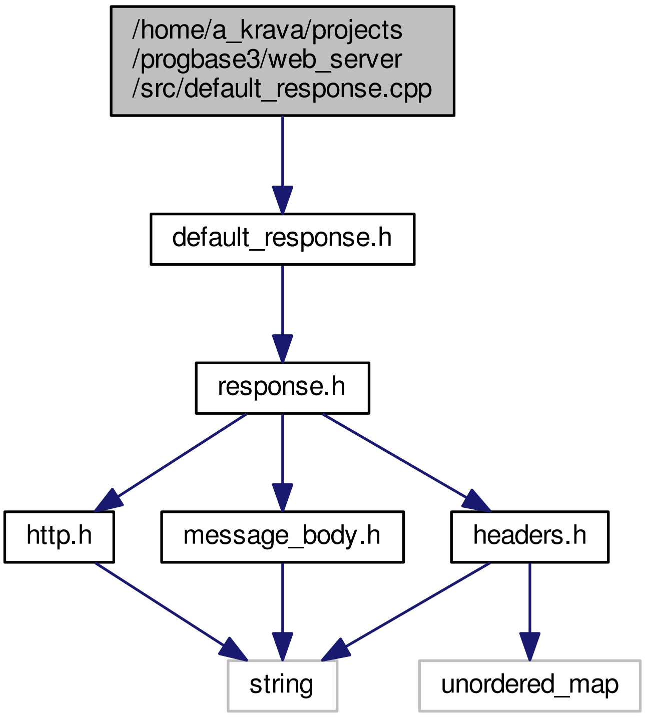
Include dependency graph for db_manager.cpp:



5.31 /home/a_krava/projects/progbase3/web_server/src/default_response.cpp File Reference

```
#include <default_response.h>
```

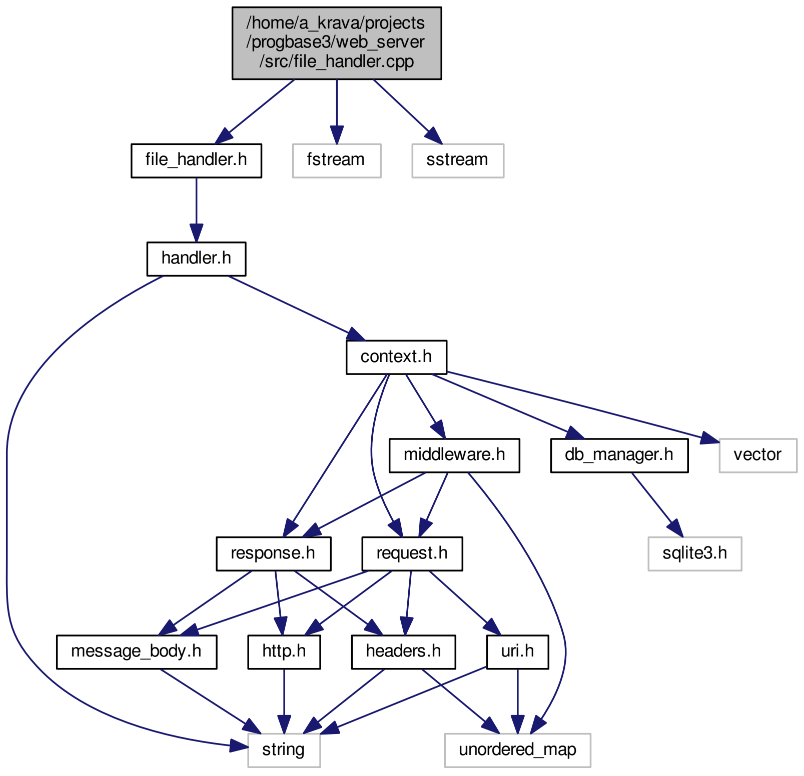
Include dependency graph for default_response.cpp:



5.32 /home/a_krava/projects/progbase3/web_server/src/file_handler.cpp File Reference

```
#include <file_handler.h>
#include <fstream>
#include <sstream>
```

Include dependency graph for file_handler.cpp:



Macros

- #define __MAX_SIZE_CACHED 5120

5.32.1 Macro Definition Documentation

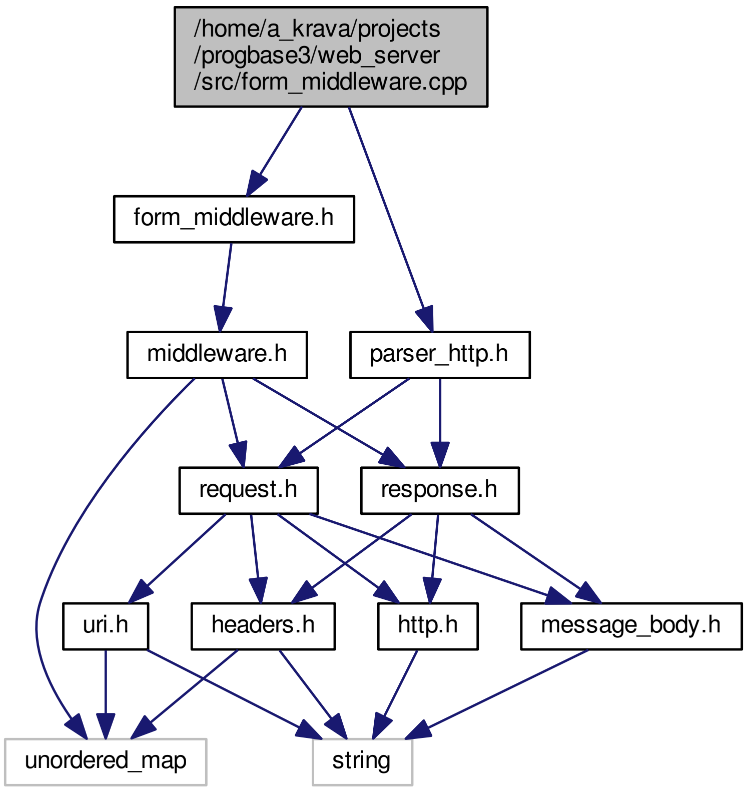
5.32.1.1 __MAX_SIZE_CACHED

```
#define __MAX_SIZE_CACHED 5120
```

5.33 /home/a_krava/projects/progbase3/web_server/src/form_middleware.cpp File Reference

```
#include <parser_http.h>
#include <form_middleware.h>
```

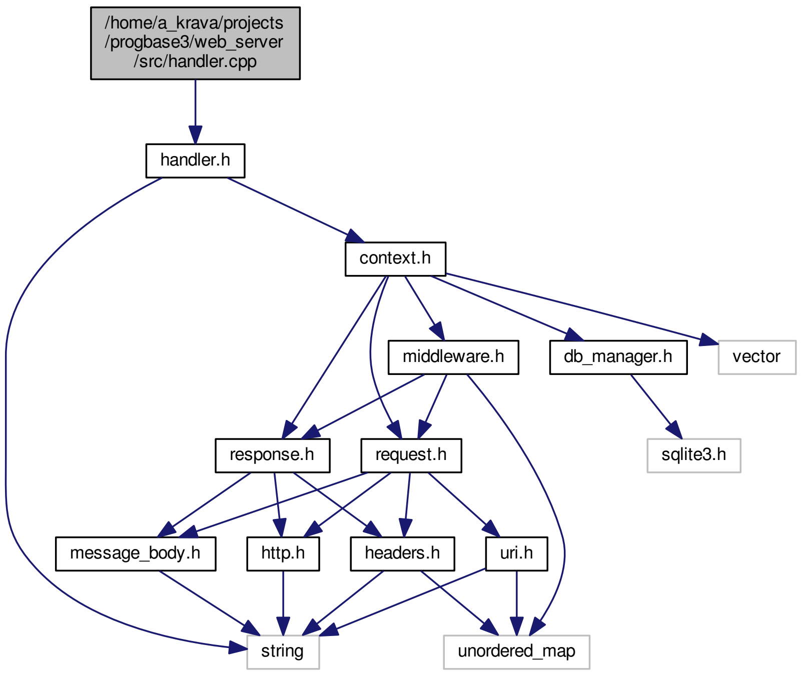
Include dependency graph for form_middleware.cpp:



5.34 /home/a_krava/projects/progbase3/web_server/src/handler.cpp File Reference

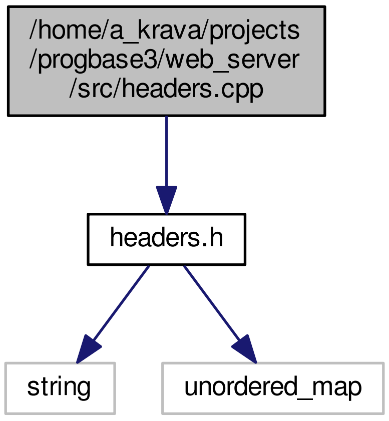
```
#include <handler.h>
```

Include dependency graph for handler.cpp:



5.35 /home/a_krava/projects/progbase3/web_server/src/headers.cpp File Reference

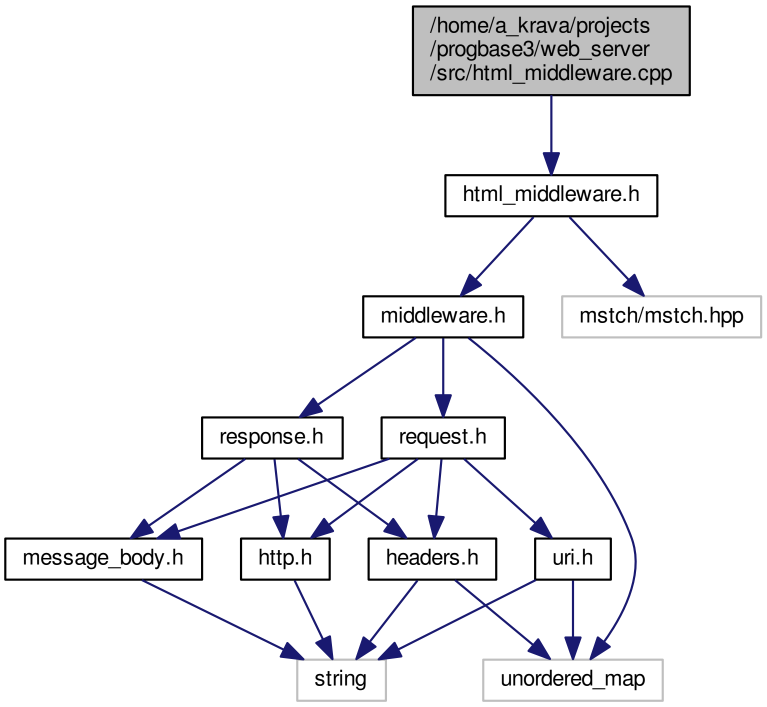
```
#include <headers.h>
```
Include dependency graph for headers.cpp:



5.36 /home/a_krava/projects/progbase3/web_server/src/html_middleware.cpp File Reference

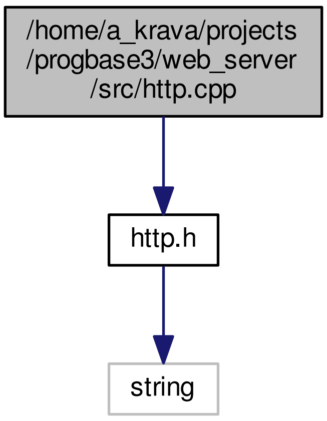
```
#include <html_middleware.h>
```

Include dependency graph for html_middleware.cpp:



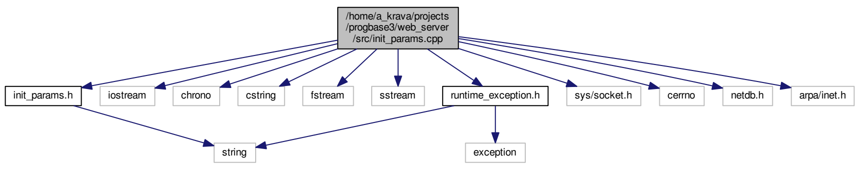
5.37 /home/a_krava/projects/progbase3/web_server/src/http.cpp File Reference

```
#include <http.h>
```
Include dependency graph for http.cpp:



5.38 /home/a_krava/projects/progbase3/web_server/src/init_params.cpp File Reference

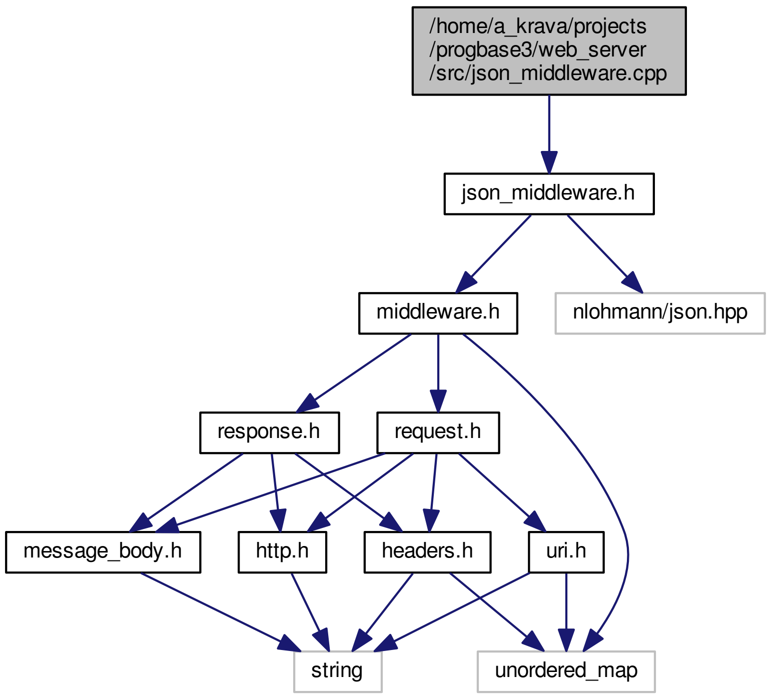
```
#include <init_params.h>
#include <iostream>
#include <chrono>
#include <cstring>
#include <fstream>
#include <sstream>
#include <runtime_exception.h>
#include <sys/socket.h>
#include <cerrno>
#include <netdb.h>
#include <arpa/inet.h>
```
Include dependency graph for init_params.cpp:



5.39 /home/a_krava/projects/progbase3/web_server/src/json_middleware.cpp File Reference

```
#include <json_middleware.h>
```

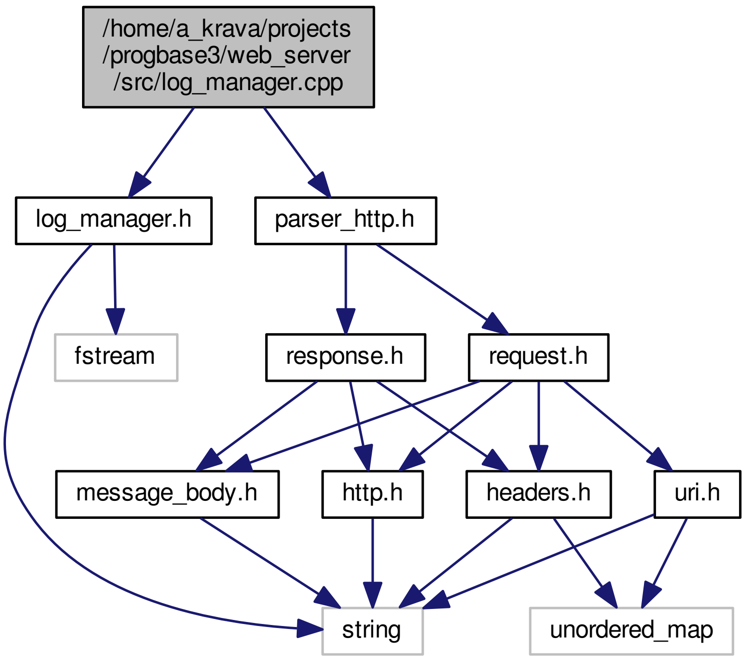
Include dependency graph for json_middleware.cpp:



5.40 /home/a_krava/projects/progbase3/web_server/src/log_manager.cpp File Reference

```
#include <log_manager.h>
#include <parser_http.h>
```

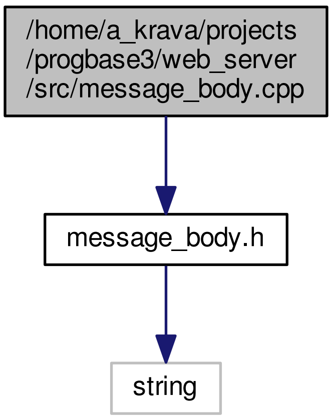
Include dependency graph for log_manager.cpp:



5.41 /home/a_krava/projects/progbase3/web_server/src/message_body.cpp File Reference

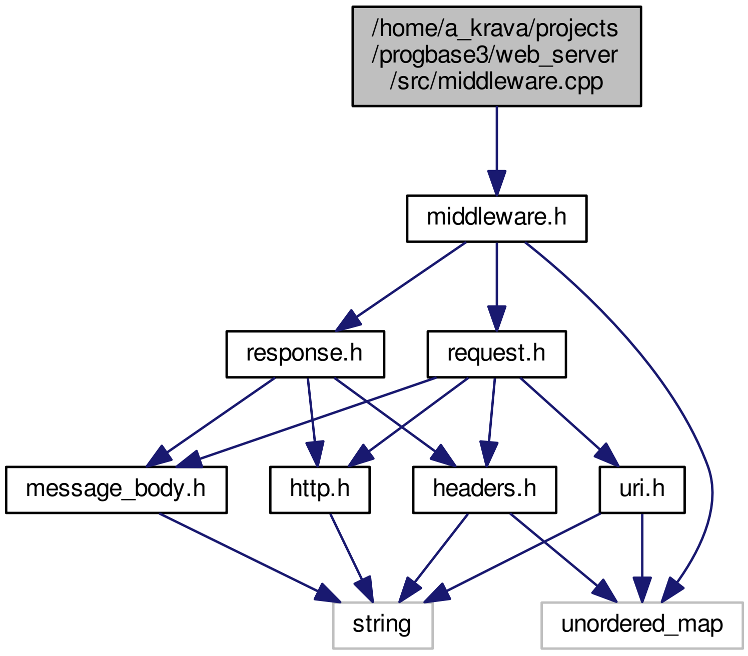
`#include <message_body.h>`

Include dependency graph for message_body.cpp:



5.42 /home/a_krava/projects/progbase3/web_server/src/middleware.cpp File Reference

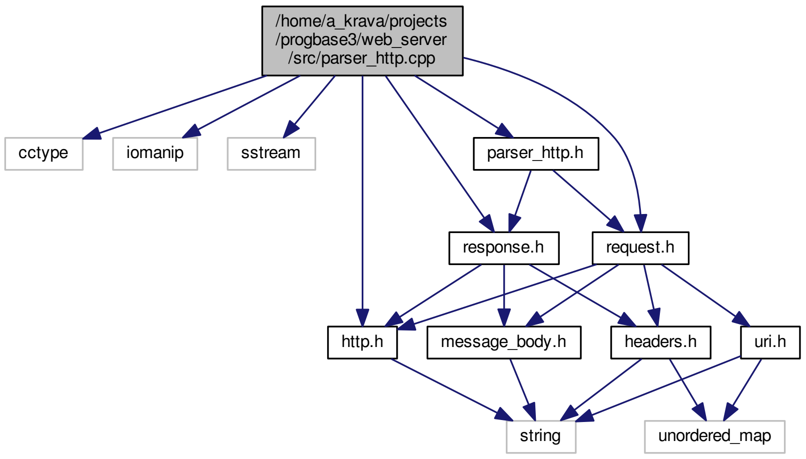
```
#include <middleware.h>
```
Include dependency graph for middleware.cpp:



5.43 /home/a_krava/projects/progbase3/web_server/src/parser_http.cpp File Reference

```
#include <cctype>
#include <iomanip>
#include <sstream>
#include <request.h>
#include <response.h>
#include <parser_http.h>
#include <http.h>
```

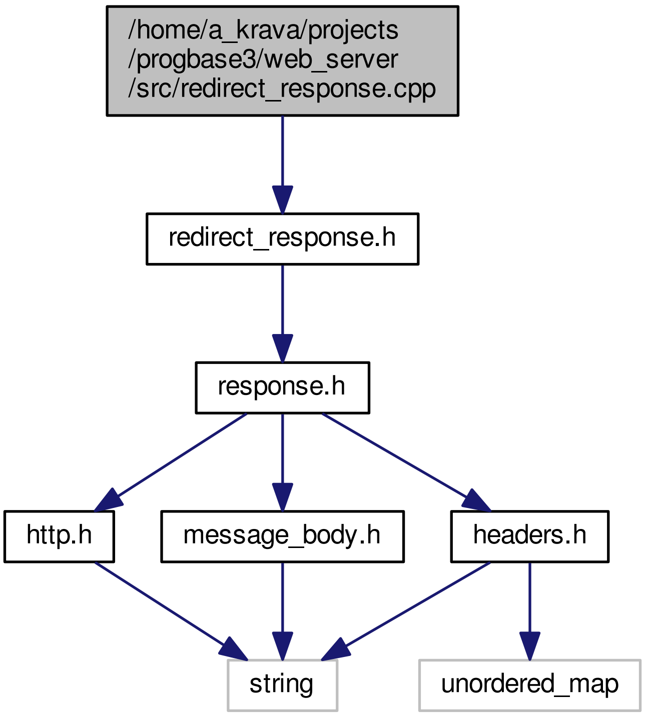
Include dependency graph for parser_http.cpp:



5.44 /home/a_krava/projects/progbase3/web_server/src/redirect_response.cpp File Reference

```
#include <redirect_response.h>
```

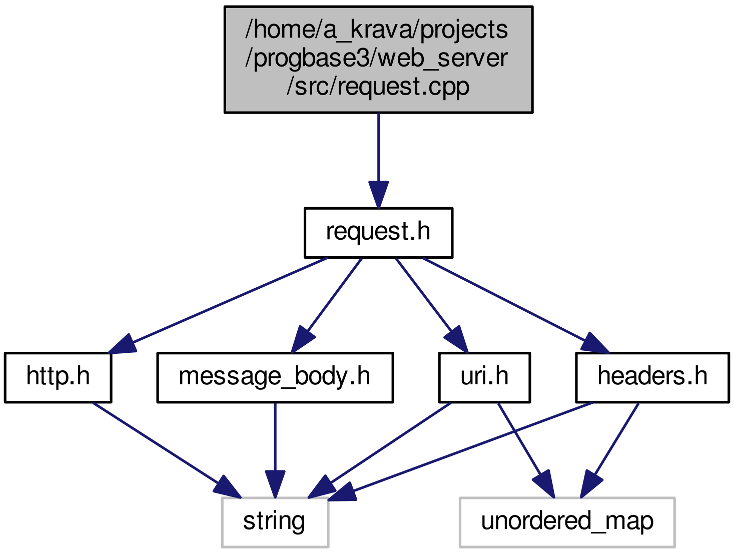
Include dependency graph for redirect_response.cpp:



5.45 /home/a_krava/projects/progbase3/web_server/src/request.cpp File Reference

```
#include <request.h>
```

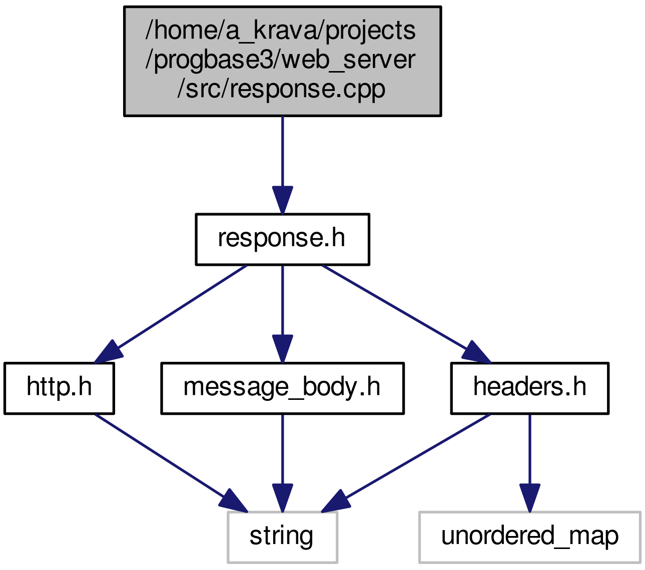
Include dependency graph for request.cpp:



5.46 /home/a_krava/projects/progbase3/web_server/src/response.cpp File Reference

```
#include <response.h>
```

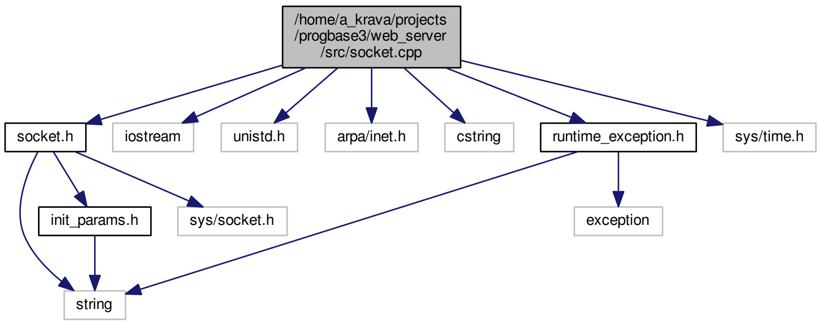
Include dependency graph for response.cpp:



5.47 /home/a_krava/projects/progbase3/web_server/src/socket.cpp File Reference

```
#include <socket.h>
#include <iostream>
#include <unistd.h>
#include <arpa/inet.h>
#include <cstring>
#include <runtime_exception.h>
#include <sys/time.h>
```

Include dependency graph for socket.cpp:



Macros

- #define __BUFFER_SIZE 1024

5.47.1 Macro Definition Documentation

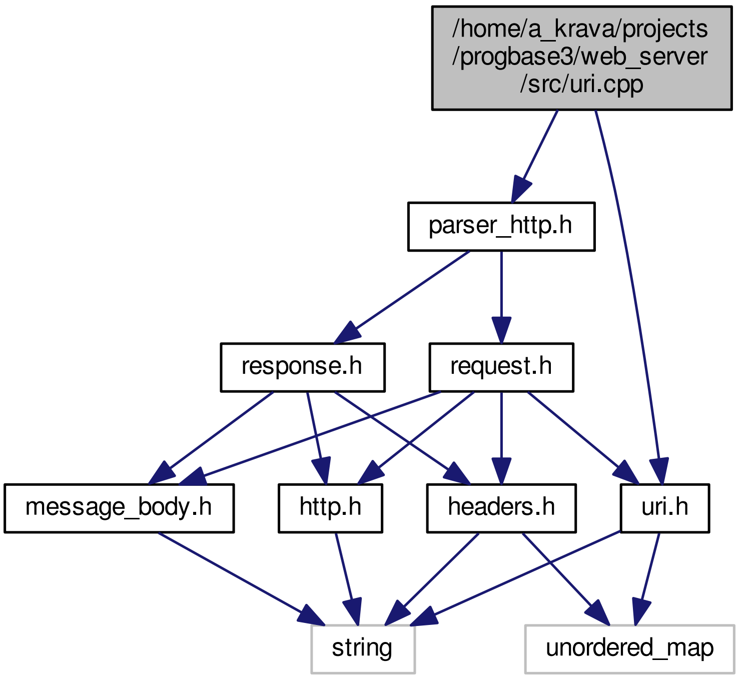
5.47.1.1 __BUFFER_SIZE

```
#define __BUFFER_SIZE 1024
```

5.48 /home/a_krava/projects/progbase3/web_server/src/uri.cpp File Reference

```
#include <parser_http.h>
#include <uri.h>
```

Include dependency graph for uri.cpp:


/home/a_krava/projects /progbase3/web_server /src/uri.cpp
parser_http.h
response.h
request.h
message_body.h
http.h
headers.h
uri.h
string
unordered_map