

AARGUL

About AARGUL

AARGUL is a Web application Archive, that can be used with web framework or with node http(s) module.

AARGUL Features

Routing

Routing refers to how an server side application responds to a client request to a particular endpoint. This endpoint consists of a URI (a path such as / or /books) and an HTTP method such as GET, POST, PUT, DELETE, etc.

The routes are stored in `[APP_DIR]/routes` directory.

Routing function

```
var app = new Aargul("/home/me/my/aargul/app");

// app.{method}(uri, handler)
app.get("/",function(req,res, next){
    res.send("<h1>Arrgul is Cool</h1>");
});
// app.{method}(uri,...middleware, handler)
app.get("/hello","sayHello",function(req,res, next){
    res.send(req.say+" world");
});
// app.{method}(uri,...middleware, handler)
app.get("/sup","cors","sayHello",function(req,res, next){
    res.send(req.say+" world");
});
// ....
```

Routing file

```
// located at [APP_DIR]/route/index.js
// curl http://127.0.0.1:3000/

function get(req, res ){
    res.send(views.render("index.njk"))
}
function post(req, res ){
    res.json(req.body); // return POST DATA
}
```

```

module.exports = {
  get,
  post
}

```

```

// located at [APP_DIR]/route/user/:login.js
// curl http://127.0.0.1:3000/users/badinga
const users = database.collection("users");
function getUser(req, res, fn){
  var user = users.find({
    login : req.params.login
  }).toArray()[0]
  if(user){
    fn(user);
  }else {
    res.status(404).send("Sorry can't find this user!");
  }
}
// get user profile
function get(req, res ){
  getUser(req, res, async (user)=>{
    res.send(views.render("users/profile.njk", {
      ...user,
      photo : await users.getFile(user.login, "data-uri")
    }));
  })
}
// update user info
function post(req, res ){
  getUser(req, res, async (user)=>{
    if(req.files.photo){
      await
users.saveFile(user.login, req.files.photo, req.files.photo, mime)
    }
    await users.updateOne({login: user.login}, {
      $set: req.body
    });
    res.send(views.render("users/profile.njk", {
      ...user,
      ...req.body
    }));
  })
}

module.exports = {
  middleware : {
    all : ["cors"],
    post : ["jsonPost", "uploadFile"],

```

```
    },  
    get,  
    post  
  }  
}
```

Middleware

Middleware in Aargul are functions that come into play after receives the request and before the response is sent to the client. They are arranged in a chain and are called in sequence.

The middleware files are stored in `[APP_DIR]/middleware` directory

Middleware function

```
var app = new Aargul("/home/me/my/aargul/app");  
  
app.middleware("sayHello", function(req, res, next){  
  req.say = "hello";  
  next();  
});  
// ....
```

Middleware file

```
// located at [APP_DIR]/middleware/sayHello.js  
  
module.exports = function(req, res, next){  
  req.say = "hello";  
  next();  
}
```

Routing	
Templating	
Static Files	
Embedded DB	

Only compatible with nodejs(v8) >= 12

AARGUL adapters:

Adapters	Implemented
Express	Yes
Connect	Yes

Adapters	Implemented
Next	wip
Koa	soon

Installation

With [npm](#) do:

```
npm install aargul
```

What is it good for?

The library allows you to:

- use archive as express router
- use archive as express router
- create archive router from directory

Dependencies

There are no other nodeJS libraries that AARGUL is dependent

Examples

Express usage

```
var express = require("express");
var Aargul = require("aargul");
var app = express();

var application = Aargul.open("./archive.gul"); // Pass file path or
buffer content

application.getInfo() // return package.json document
// mount router
app.use("/", application);
/// throw
///   RangeError if the application the hash value is not correct
///   TypeError if the application is not an archive
///   EvalError if the application dependencies is not found

// mount router if dependencies is satisfy
if(application.canMount){
```

```

    app.use("/", application.express);
  }else {
    console.error("MISSING")
    console.table(application.missingDependencies) // return all
    missing dependencies
  }

```

Creating a archive usage

```

// creating an archives programmatically

/** Dir structure
 *
 * [dirname]/views/name.njk -- View
https://mozilla.github.io/nunjucks/getting-started.html
 * [dirname]/views/name/cool.njk --
Viewhttps://mozilla.github.io/nunjucks/getting-started.html
 * [dirname]/routes/name.js -- name is string -
Eg. user, book or the special value index
 * [dirname]/routes/:userId/books/:bookId.js -- see
https://expressjs.com/en/guide/routing.html
 * -- Eg.
users/[id].js, page[num].js
 * [dirname]/middlewares/*.js -- name is static
 * [dirname]/assets/* -- this is the
static directory accessible via /assets,
 * -- the url "/assets"
can be change in package.json "assetsRootUri":"/newName", Eg: "/public"
 * [dirname]/package.json -- the information
about application
 *
 * **/

// add router application
var app = new Aargul("/home/me/my/aargul/app");

// ADD Manualy
//// Middleware
app.middleware("sayHelloMiddleware", function(req,res, next){
  req.say = "hello";
  next();
});

//// ADD route
/// Method Allowed : all, get, post, put, head, delete, options, trace,
copy, lock, mkcol, move, purge, propfind, proppatch, unlock, report,
mkactivity, checkout, merge, "m-search", notify, subscribe, unsubscribe,
patch, search, and connect
app.get("/hello", "sayHelloMiddleware", function(req,res){

```

```

        res.end(req.say + " world")
    });

    ///// ADD assets content
    var content = "inner content of the file";
    app.assets("test.txt", {
        content : Buffer.from(content, "utf8"), //file content, default
        undefined
        comment : "entry comment goes here", // comment when list directory,
        default undefined
        replace : true, // replace if path exists if not defined replace is
        true for file and false for directory
        publicUri : "super.txt" // under assets directory if not defined
        basename is used
    });
    /// DEFAULT OPTIONS IS
    /* {
        comment : null
        replace : false,
        publicUri : "some_picture.png" // path is set from the basename
        : accessible via /assets/some_picture.png
    }
    */
    app.assets("/home/me/some_picture.png");
    app.assets("/home/me/special_picture.png", {publicUri : "pic.png"});
    // add directory
    app.asset("/home/me/assets"); // add assets directory // merge if
    existing
    app.asset("/home/me/js",{publicUri : "script"}); // add "assets/js"
    directory // merge if existing
    app.asset("/home/me/public",{replace:true}); // add static directory
    accessible via "/assets/public" (replace if existing)
    app.asset("/home/me/public",{replace:true, publicUri : "/js"}); // add
    static directory accessible via "/assets/js" (replace if existing)

    // get everything as a buffer
    var willSendthis = app.toBuffer();
    // or write everything to disk
    app.save(/*target file name*/ "/home/me/archive.gul");

```

build unknown