

MixerJS

Software Requirements Specification

v1

19 June 2015

INTRODUCTION

Every time a website gets loaded, additional dependency files need to be loaded as well, It can be images, javascript files or css files. For each dependency the server open a HTTP connection to the server. If there are many dependencies, this can reduce the performance of the page in a great manner. Many times the browser need to wait until all dependencies are loaded, reducing the user experience.

For optimizing a web performance there are three main types of optimization techniques:

1. Reducing the amount of HTTP requests
2. Reducing the overall content of the file requested.
3. Techniques that promote parallelization

Reducing the number of request made by the client can be done by concatenating javascript or css files. Each request takes at least 20 ms, that it can seem not to much but if the requests grow, performance on the page starts to decrease.

Reducing the content of each file can be done by eliminating unused space and for javascript files, renaming variables, removing comments. Javascript files can be reduced by more than 80% on size, css 25%.

Reference :
<http://www.yottaa.com/blog/application-optimization/bid/259514/How-Does-Reducing-JavaScript-Requests-Minifying-JavaScript->

Normally these procedures are done during deployment of your website, but it takes time to set up and knowledge to use the specific tools.

MixerJS

Its a web server that receives GET requests with configuration parameters and it serves multiple JavaScript or CSS libraries in a single request.

Specifications

Folder Structure

- models/
- controllers/
- builds/

- mixer.js
- README.md
- config.js
- package.json
- .gitignore

Code Style

The code must be validated using the [style guide](#)

Library Dependency Logic

- In case there is a dependency conflict between version of library, choose the most updated one

Possible Future Features

- Be able to compile different languages like, JADE, SASS, LESS, Coffescript, etc.
- Use local files into the response.
- Client side tool for creating the urls

Error Management

Error management should be done through a middleware

Functional Tests

Tests must be done in the node server and in the browser.

Successful Cases

- Single library
/mixer.js?jquery
/mixer.js/intense-images
/mixer.js/angular

/mixer.css/foundation

/mixer.css/bootstrap

- Minified library

/mixer.min.js?hammerjs

/mixer.min.js?epoch

/mixer.min.js?wow

/mixer.min.css/fontawesome

/mixer.min.css/pure

- Library with version

/mixer.js?blast=0.1.0

/mixer.js?visjs=3.11.0

/mixer.css?fontawesome=3.2.0

/mixer.css?bootstrap=3.11.0

- Multiple libraries

/mixer.js?lodash&jquery&angular

/mixer.js?mathjs=1.1.0&jquery&react=0.3.3

/mixer.js?lodash=2.1.0&angular=1.1.3

/mixer.css?fontawesome=3.2.0&pure=0.2.1&fontawesome

/mixer.css?bootstrap=3.11.0&foundation

- Libraries with dependencies, using automatic dependency

/mixer.js?backbone&wow&ember

/mixer.js?toastr&jquery

/mixer.css?ionic=1.0.0&pure=0.2.1

/mixer.css?bootswatch=2.0.4&pure

- Custom build for libraries

/mixer.js?lodash={"category":["array"]}

/mixer.js?lodash={"category":["array"], "plus": ["random", "template"]}

/mixer.js?lodash={"type":"modern", "include": ["random", "template"]}

/mixer.js?lodash={"exports":"amd", "category":["collection"], "plus": ["random", "template"], minus: ["shuffle"]}

/mixer.min.js?jquery={"exclude": ["ajax"]}

/mixer.js?jquery={"exclude": ["css", "effects"]}

Unsuccessful Cases

- Same library
/mixer.js?jquery&jquery
- Invalid library
/mixer.js?jqueryy
- Invalid version
/mixer.min.js?jquery=9.6
- Library with dependencies, not using automatic dependency
/mixer.js?toastr
- Invalid params for custom builds
/mixer.js?lodash={"category":["array", 'invalid']}

Errors

- **notFound**
Occurs when a library is not found or it's version

```
GET /mixer.js?jqueryy
Response 404 (application/json)
{
  "error": "notFound",
  "message": "jquery version 1.5 not found"
}
```

- **conflictLibrary**
Same library in the query

```
GET /mixer.js?jquery&jquery
Response 409 (application/json)
{
  "error": "conflictLibrary",
  "message": "Duplicated libraries in the query are not allowed.Check jquery library"
}
```

- **dependencyNeeded**

GET /mixer.js?toastr

Response 412 (application/json)

```
{
  "error": "dependencyNeeded",
  "message": "jquery, angular are dependencies of toastr"
}
```

- **libraryNoStandard**

- **unknown**

Response 400 (application/json)

```
{
  "error": "unknown",
  "message": "An error occurred, please contact support@jortech.com.ve"
}
```

- **Custom**

Any other type of error that can be added through custom builds