

Class 14 (Middleware)

```
node_modules/
package.json
server.js
    app.get('/articles/find', to locate a specific instance of an article record)
    app.get('/articles/categories', gathers all unique categories from the articles table)
public/
    admin.html
    index.html
    new.html
    data/hackerIpsum.json
    scripts/
        controllers/
            aboutController.js
                aboutController.index()
            articleController.js
                articleController.index(ctx)
                articleController.loadById = (ctx, next) =>
                articleController.loadByAuthor = (ctx, next) =>
                articleController.loadByCategory = (ctx, next) =>
                articleController.loadAll = (ctx, next) =>
            routes.js
                page('/', articleController.index)
                page('/about', aboutController.index)
                page('/article/:article_id', articleController.loadById, articleController.index);
                page('/category', '/');
                page('/author', '/');
                page('/author/:authorName', articleController.loadByAuthor,
                    articleController.index);
                page('/category/:categoryName', articleController.loadByCategory,
                    articleController.index);
                page()
        models/
            article.js
                Article()
                Article.all = []
                Article.prototype.toHtml() [this has been relocated the the view]
                Article.loadAll()
                Article.fetchAll()
                Article.findWhere() [queries DB to GET a specific record]
                Article.allCategories() [uses a GET to gather all of the categories]
                Article.numWordsAll()
                Article.allAuthors()
                Article.numWordsByAuthor()
                Article.stats()
                Article.truncateTable()
                Article.prototype.insertRecord()
                Article.prototype.deleteRecord()
                Article.prototype.updateRecord()
            repo.js
                const repos = {};
                repos.all = [];
                repos.requestRepos() [TODO: this will require a refactored AJAX call]
                repos.with()
        views/
            adminView.js
                adminView.initAdminPage()
            articleView.js
                render() [relocated version of the Article.prototype.toHTML() method]
                articleView.populateFilters()
                articleView.handleFilters() [combines the filter handling into one method]
                articleView.handleAuthorFilter() [deleted]
                articleView.handleCategoryFilter() [deleted]
                articleView.setTeasers() [deleted and replaced with a link in article template]
                articleView.index() [refactored version of articleView.initIndexPage()]
            newArticle.js
                newArticle.initNewArticlePage()
                newArticle.create()
                newArticle.submit()
            repoView.js
                const repoView = {}
                ui()
                repoView.index()
```